NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

**3D BATTLESPACE VISUALIZATION
USING OPERATIONAL PLANNING DATA**

by

Claude O. Hutton, Jr.

September 2003

Thesis Advisor:	Don Brutzman
Co-Advisor:	Curtis Blais

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: 3D Battlespace Visualization Using Operational Planning Data

5. FUNDING NUMBERS

6. AUTHOR Claude O. Hutton, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NSA, Ft. Meade, MD
SPAWAR, San Diego, California

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT

In modern combat operations today the display of operational data is still tied to stove-piped and proprietary systems and software. Additionally, combat systems are still using 2D displays of the battlefield in order to reflect a picture of the battlefield to the warfighter. Stepping away from stove-piped and proprietary systems and reflecting a 3D picture of the battlefield is the direction that this thesis research explores.

Research is conducted to explore technologies needed to provide operational forces with web-based 3D visualizations of operational data. Technologies used in this research are XML, XSLT, JAVA, X3D and VRML. A prototype application is developed that allows for the 3D display of operational data. The research demonstrates how operational data can be stored in a database and accessed through a web-based 3D representation of the area of operation. Data sets used in this prototype include Digital Terrain Elevation Data and operational planning data. Access to the data is provided through a web-based interface. The web-based view of the data provides both 2D and 3D views. This research shows that current open source technology can provide the warfighter with a web-based 3D view of the battlefield.

14. SUBJECT TERMS
Virtual Environments, Extensible 3D Graphics, X3D, Extensible Markup Language, XML, Java, Scenario Generation, Combined Operational Picture, COP, Extensible Modeling and Simulation Framework (XMSF), SAVAGE, database

15. NUMBER OF PAGES
126

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

3D BATTLESPACE VISUALIZATION
USING OPERATIONAL PLANNING DATA

Claude O. Hutton, Jr.
Captain, United States Marine Corps
B.S., Old Dominion University, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author: Claude O. Hutton, Jr.

Approved by: Don Brutzman
Thesis Advisor

Curtis Blais
Co-Advisor

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In modern combat operations today, the display of operational data is still tied to stove-piped and proprietary systems and software. Additionally, combat systems are still using 2D displays of the battlefield in order to reflect a picture of the battlefield to the warfighter. Stepping away from stove-piped and proprietary systems and reflecting a 3D picture of the battlefield is the direction that this thesis research explores.

Research is conducted to explore technologies needed to provide operational forces with web-based 3D visualizations of operational data. Technologies used in this research are Extensible Mark-up Language (XML), Extensible Stylesheet Language for Transformation (XSLT), JAVA, Extensible 3D Graphics (X3D) and Virtual Reality Modeling Language (VRML). A prototype application is developed that allows for the 3D display of operational data. The research demonstrates how operational data can be stored in a database and accessed through a web-based 3D representation of the area of operation. Data sets used in this prototype include Digital Terrain Elevation Data and operational planning data. Access to the data is provided through a web-based interface. The web-based view of the data provides both 2D and 3D views. This research shows that current open source technology can provide the warfighter with a web-based 3D view of the battlefield.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION..1
A. PROBLEM STATEMENT 1
B. OVERVIEW..3
C. MOTIVATION 3
1. My Interest and Background 4
2. The Need for 3D Visualization of Data in Military Operations.......5
D. OBJECTIVES 5
E. THESIS ORGANIZATION...6

II. RELATED WORK 7
A. INTRODUCTION...7
B. RELEVANT WEB TECHNOLOGIES 7
1. XML 7
2. XSLT...8
3. JAVA...8
4. Extensible 3D (X3D) Graphics...8
C. RELATED RESEARCH 9
1. Major Shane Nicklaus' Thesis: XML OPORDER VISUALIZATION 9
2. LT James Harney's Thesis: AGENT-BASED ANTI-TERRORISM/FORCE PROTECTION (AT/FP) MODEL.............9
D. SUMMARY 9

III. DEVELOPMENT OF 3D SCENES, DATABASE AND EXEMPLAR................11
A. INTRODUCTION...11
B. DEVELOPMENT TOOL DESCRIPTION 11
1. X3D-Edit..11
2. VRMLPad 2.0..12
3. Microsoft's WordPad 12
4. XML SPY...12
5. Stylus Studio...12
6. Polytrans for Windows..12
7. Internet Model Optimizer 13
8. MultiGen Creator V2.5 13
9. Microsoft's SQl Server 13
10. Tomcat Server...13
11. VRML Devicehandler by Geovrml.com 14
12. Microsoft's Internet Information Services (IIS) 14
13. Internet Explorer 14
14. Cortona VRML Client 4.1..14
C. DEVELOPMENT OF A 3D MODEL FOR URBAN OPERATIONS......14
1. Overview 14

2. **Naval Postgraduate School Campus Model 15**
   - *a. The Old Campus Model 15*
   - *b. Reasons to Build a New Model..18*
   - *c. Establishing the Foundation/Terrain 20*
   - *d. Placing Buildings and Resizing...22*
   - *e. Adding a Fence 23*
   - *f. Adding a New Herrmann Hall 24*
   - *g. Resizing of Textures to a Power of Two................................24*
   - *h. Current Status of Model 25*

3. **Baghdad Model 25**
   - *a. The Original Model..25*
   - *b. Conversion of Model from OpenFlight Format Versus the VRML Format..26*
   - *c. Conversion Process 27*
   - *d. Current Status of Model 28*

4. **Recommendations and Conclusions 29**
   - *a. Overview 29*
   - *b. Recommendations for Further Development of the Campus Model..29*
   - *c. Recommendations for Further Development of the Baghdad Model 29*
   - *d. Conclusions 30*

D. DEVELOPMENT OF A 3D MODEL FOR TERRAIN WITHIN THE AREA OF OPERATION (AO)...30

1. **Overview 30**
2. **Island of Oahu 30**
   - *a. Obtaining the DTED Data...30*
   - *b. Elevation Grids Versus Indexed Face Set Terrain................31*
   - *c. Mesh Versus Delaunay Terrain 32*
   - *d. Building the Terrain File...34*
   - *e. Building the Texture for the Terrain 34*
   - *f. Overlaying Texture Maps on the Terrain 35*
   - *g. Modifying the Terrain..36*
   - *h. Conversion to VRML Format...37*
   - *i. Inline of Terrain into X3D..37*
   - *j. Placing Objects on the Terrain...39*
   - *k. Using Oahu Model for a Non-Combatant Evacuation Operation (NEO)...39*
   - *l. Use of Ships, Aircraft, and Other Military Models................41*
   - *m. Use of Waypoint Interpolator Prototype 42*
   - *n. Use of View Position Orientation Prototype 43*
   - *o. Conversion of Oahu Model to X3D Geospatial Profile.........44*
   - *p. Current Status of Model 44*

3. **Country of Iraq 45**
   - *a. Obtaining DTED Data...45*

b. Building the Terrain File 45
c. Using a Different Map Projection 45
d. Tiling Terrain 45
e. Building the Textures for the Terrain 46
f. Overlaying Texture Maps on the Terrain Tiles 46
g. Conversion to VRML Format 47
h. Conversion of Model to X3D Geospatial Profile 47
i. Inline of Terrain Tiles in X3D Geospatial Profile 48
j. Placing Objects on the Terrain 48
k. Using Joystick for Navigation 49
l. Development of Aircraft for Realistic Flight 49
m. Use of X3D Geospatial Profile Node Inline 51
n. Current Status of Model 51
4. Recommendations and Conclusions 52
a. Overview 52
b. Recommendations for Further Development of the Oahu Model 52
c. Recommendations for Further Development of the Iraq Model 53
d. Conclusions 53
E. 3D MODELING OF RELATIONAL DATA RETRIEVED FROM DATABASES 53
1. Choosing a Database 53
2. Setting Up SQL 2000 for XML Support 54
3. Testing XML Support on Database 54
4. Operational Data Used in This Project 54
5. Conversion of Data for the Project 54
6. Setting Up Tables in the Database 55
7. Stored Procedures and Queries Used in the Database 55
F. DEVELOPMENT OF WEB APPLICATION FOR DATA DISPLAY IN A 3D MODEL 55
1. Overview and System Architecture 55
2. Web Site Structure 56
3. Setting Up the Tomcat Server 56
4. Relationship Between the Database, XML, X3D, VRML and XSLT 56
5. XSLT Design 56
6. Development of VRML Prototypes 57
7. Java Servlets and Classes Used in Application 57
8. Use of VB Script in Web Pages for Latitude and Longitude Conversion 58
G. SUMMARY 58

IV. DEVELOPMENT RESULTS 59
A. INTRODUCTION 59
B. TESTING AND ASSESSMENT 59

C. FUNCTIONALITY OF THE 3D MODELS FOR URBAN OPERATIONS 59
1. Overview 59
2. Navigation 60
3. Frame Rate Testing 61
4. Scaling and Placement of Objects 62
5. Texture Appearance 62
6. Comparison of 3D Model with Real Perspective 62
7. Time Versus Appearance 63
D. FUNCTIONALITY OF THE 3D MODEL FOR TERRAIN WITHIN THE AREA OF OPERATION (AO) 64
1. Overview 64
2. Navigation 64
3. Frame Rate Testing 64
4. Scaling and Placement of Objects 64
5. Map Texture Quality for Terrain 65
6. Time Versus Appearance 65
E. RENDERING OF DATA WITHIN A 3D MODEL 66
1. Design of Database Tables 66
2. Output of XML from Database 66
3. Use of Java and XSLT 67
4. Development of VRML Prototypes 67
F. WEB SITE, INTERACTION WITH THE DATABASE, AND RENDERING OF 3D MODELS 68
1. Layout and Development of Web Site 68
2. Utilization of Apache Tomcat HTTP Server 68
G. SUMMARY 68

V. RECOMMENDATIONS AND CONCLUSIONS 69
A. TECHNICAL FINDINGS 69
1. Overview 69
2. Visualization of Data 69
3. Development of Urban Models 69
4. Development of Terrain 70
5. Functionality of Web Application 70
B. BENEFITS 71
1. Overview 71
2. Urban Terrain 71
3. Modeling Terrain for an Area of Operation 71
4. Course of Action Development 72
5. Consolidating, Reviewing and Analyzing Operation Data 72
6. Open Source Development 72
C. RECOMMENDATIONS 72
1. Overview 72
2. Future Development 73
a. Conversion of Entire DTED Library 73

b. Scalable Vector Graphics (SVG) 73
c. Continued Development with X3D Geospatial Profile 73
d. Navigation Aids 74
e. Prototype Development and MilStd 2525b 74
f. Near-Real-Time Control of Remote Targeting Via Network Protocols 74
g. Conversion of Web-based Application to Web Service Application 74
3. Scalable 3D Combined Operational Picture (3D COP) of the Battlefield 75
a. Phase One: Replicating the Current 2D COP in a 3D COP Battlespace 75
b. Phase Two: Exposing Non-traditional Data Sources Via a Web Services Architecture 75
c. Phase Three: Leverages the Integration of Artificial Intelligence (AI) and Agent Based Tools 76
4. Conclusion 76

APPENDIX A. ACRONYMNS AND ABBREVIATIONS 77

APPENDIX B. X3D GEOSPATIAL PROFILE CODE USED IN TERRAIN TILES 79

APPENDIX C. X3D CODE FOR FLIGHT CONTROLS USD TO CONTROL V22 IN IRAQ SCENE 81

APPENDIX D. MAIN STORED PROCEDURE USED IN PROJECT 89

APPENDIX E. XSLT TO CONVERT XML QUERY RESULTS FILE INTO X3D FORMAT 93

APPENDIX F. X3D PROTOTYPE CODE FOR ICOSN THAT REPRESENT THE DATA FROM THE RELATIONAL DATABASE 97

LIST OF REFERENCES 103

INITIAL DISTRIBUTION LIST 105

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Screen capture taken from a command and control system displaying friendly and enemy units on a map in a 2D display. [Koester 2003]................1
Figure 2. Screen capture taken from 3D model of Iraq showing operational data from a relational database being displayed in a 3D scene. 2
Figure 3. Simple example of an XML document...8
Figure 4. Map of NPS campus published by the school in February, 1995...................15
Figure 5. Screen shot of the original 3D campus model in X3D. 16
Figure 6. Picture taken from top of Spanagel Hall looking toward Root Hall on July 28, 2003..17
Figure 7. Old NPS Campus model showing same position as photo in Figure 6. 17
Figure 8. Screen shot of old campus model showing the spacing and location of two buildings..19
Figure 9. Campus map showing the spacing and location of the same two buildings shown in Figure 8..19
Figure 10. Airplane used to take aerial photo of campus...20
Figure 11. Original photo taken of campus in the year 2002..21
Figure 12. Image of campus after modifications to the photo in Figure 11.....................22
Figure 13. NPS Campus model with an X, Y, Z axes..23
Figure 14. Inside screen capture of the Herrmann Hall Model.......................................24
Figure 15. Screen shot of new NPS campus model with 3D buildings placed on top of a geo-referenced image-map photograph..25
Figure 16. Screen shot of Baghdad VRML model by the contractor Object Raku through the US ARMY Research, Development and Engineering Command, Simulation Technology Center..26
Figure 17. Baghdad model showing popup selection for different times of day and weather conditions. 27
Figure 18. Screen shot from MultiGen Creator tool showing item grouping. 28
Figure 19. Screen shot of new Baghdad VRML model currently being developed. 29
Figure 20. MultiGen Creator screen shot showing ability to read DTED data................31
Figure 21. Screen shot showing polygons of the Oahu terrain using a Delaunay terrain system...32
Figure 22. Screen shot showing polygons of Oahu terrain using a polymesh terrain system. 33
Figure 23. Screen shot showing polygons of a test terrain file built from an elevation grid. 33
Figure 24. MultiGen Creator window showing the terrain building options...................34
Figure 25. Completed image texture built from several 1:50,000 map sheets of Oahu....35
Figure 26. Screen shot showing Oahu map texture being mapped onto polygons in MultiGen Creator. 36
Figure 27. Modified Oahu scene showing polygons removed from the harbor and Ford Island displayed in MultiGen Creator. 37

Figure 28. Screen shot of X3D–Edit tool showing import feature....................................38
Figure 29. Final Oahu scene with axis. 39
Figure 30. Screen shot from Oahu scene showing use of domes to represent area of interest...40
Figure 31. Screen shot from NPS campus in Oahu scene. 41
Figure 32. Screen shot of military models used in Oahu scene. 42
Figure 33. Simulated Non-Combatant Evacuation Operation in Oahu scene demonstrating use of waypoint interpolator to move aircraft.........................43
Figure 34. Screen shot showing an example of the View Position Orientation prototype. 44
Figure 35. Iraq Scene showing curved terrain...45
Figure 36. Screen shot of one of the 25 tiles used in the Iraq scene. Each tile represented an area 2 degree in latitude by 2 degree in longitude..................46
Figure 37. Iraq scene showing the 25 individual tiles together.......................................47
Figure 38. Iraq scene showing placement of WASP class ship using geo-location prototype. 48
Figure 39. V22 aircraft model in flight using joystick to control in the Iraq scene. 50
Figure 40. Catalina model developed by David "Fuzzy" Wells, USAF. 51
Figure 41. Screen shot taken from the northern area of Iraq looking southwest. This screen shot shows the mountain ranges in the northeast of Iraq.....................52
Figure 42. Herrmann Hall scene showing user stuck in scene while in walk navigation mode. The floor can be scene overhead.......................................61
Figure 43. Comparison of campus photos and 3D models...63

LIST OF TABLES

Table 1. Frame rates for urban scenes..62
Table 2. Frame rates for terrain scenes. 64

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to acknowledge the direct financial support of the Space and Naval Warfare Systems Center, San Diego California (SSC San Diego), and the National Security Agency’s Cryptological Lab at, Monterey California for sponsoring portions of this research. The author would also like to thank the following list of people whose assistance, support, and contributions made this thesis possible:

- Don Brutzman and Curt Blais, from the Naval Postgraduate School (NPS) for their guidance and support during the course of this thesis research.

- Jeff Weekley from NPS for providing general assistance with learning both 2D and 3D graphics and providing useful feedback during the course of this research.

- Matt Prichard and John Locke from NPS who taught me how to use the Creator tool and for building several of the models.

- George Zolla and my other instructors who taught me at NPS and provided me with the tools necessary to complete this thesis.

- My fellow students (James Neushul, Sean Hynes, James Harney, Steve Simmons and the entire SAVAGE group) who helped out with coding and provided encouragement and great insight into the possibilities for the future of the warfighter.

- My fellow Marines who fought in Operation Iraqi Freedom while research for this thesis was conducted.

- My wife Allison, my sons William and Andrew and my daughter Sophie for their encouragement, tolerance, and perseverance during my late nights at school. With out their support and love this would not have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

If you want peace, prepare for war. Si vis pacem para bellum.
--Roman maxim circa 5th Century A.D.

A. PROBLEM STATEMENT

In modern combat operations today the display of operational data is still tied to stove-piped and proprietary systems and software. Examples of stove-piped and proprietary systems would be: Global Command and Control System (GCCS), Theater Battle Management Control System (TBMCS), and Intelligence Analysis System (IAS). Additionally, combat systems are still using two dimensional (2D) displays of the battlefield in order to reflect a picture of the battlefield to the warfighter (Figure 1). Stepping away from stove-piped and proprietary systems and reflecting a three dimensional (3D) picture of the battlefield is the direction that this thesis research explores.

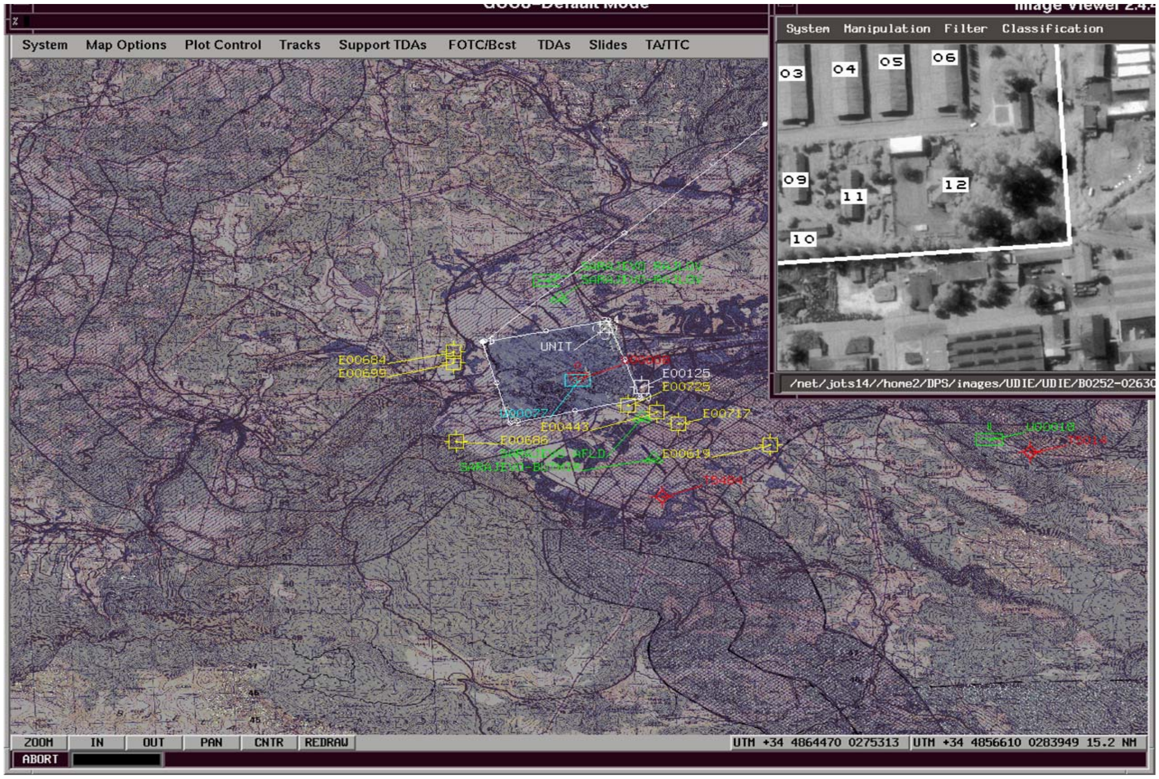


Figure 1. Screen capture taken from a command and control system displaying friendly and enemy units on a map in a 2D display. [Koester 2003]

This thesis presents the investigation, research and development of a web-based exemplar for displaying operational data in a 3D scene. This project demonstrates how operational data such as Digital Terrain Elevation Data (DTED) [DTED 1996] and data from a relational database can be combined and displayed in a 3D scene through a web browser (Figure 2). Technologies used in this research are XML, XSLT, JAVA, X3D and VRML. This research demonstrates how operational data can be stored in a database and accessed through a web-based 3D representation of an area of operation. The web-based view of the data provides both 2D and 3D views of the data. This research shows that current open source technology we can provide the warfighter with a web-based 3D view of the battlefield. An assumption is made in this thesis that a 3D view of the battlespace enhances the ability of commanders and their staff to plan and execute military operations. Follow-on research can be conducted to explore the validity of this assumption.

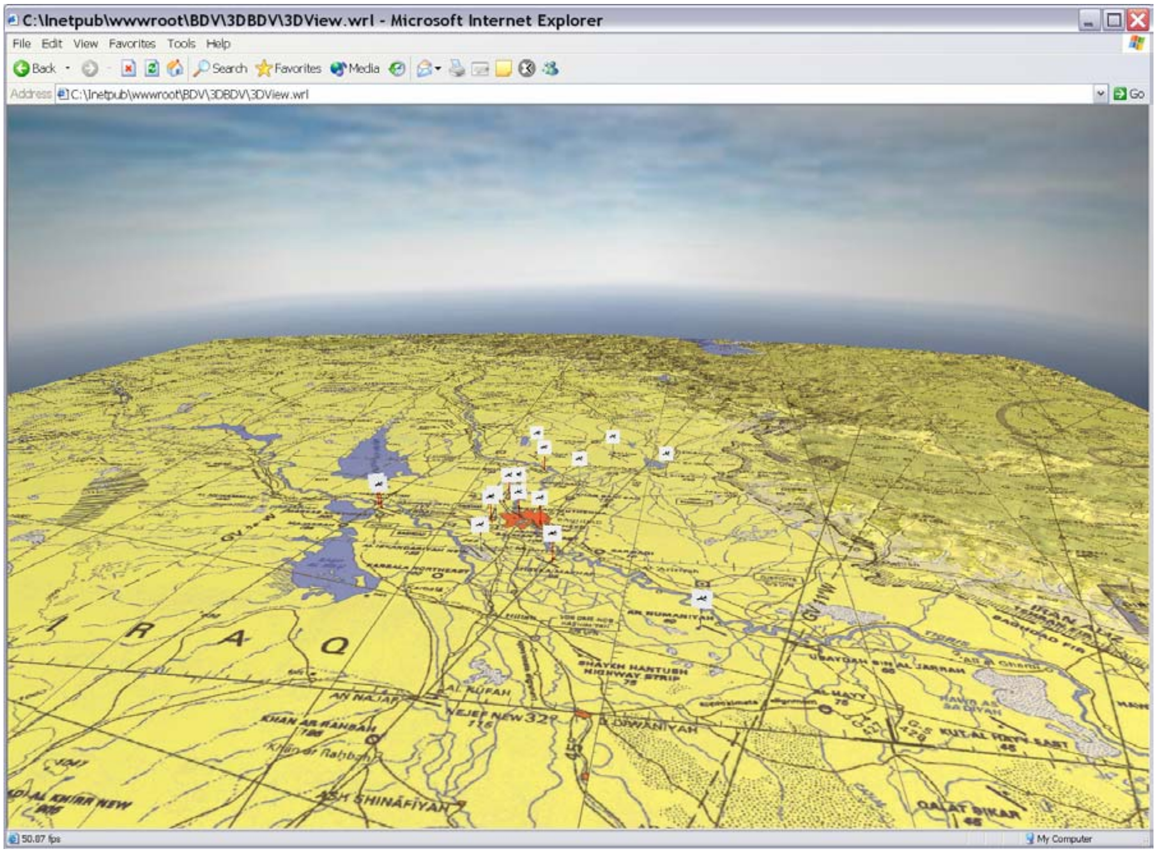


Figure 2. Screen capture taken from 3D model of Iraq showing operational data from a relational database being displayed in a 3D scene.

B. OVERVIEW

In military operations a commander must maintain a perspective of the battlefield at all times. In planning for combat operations analysis of the mission, terrain, enemy and weather must be accomplished prior to committing forces. During combat operations the commander and his staff must maintain a perspective of the battlefield in conjunction with enormous amounts of data that can come from multiple sources and in most cases are on several different systems.

During the planning phase of combat operations a Military Combined Obstacle Overlay (MCOO) is prepared by the intelligence staff for the commander and his staff. As an intelligence officer the author of this thesis has prepared several MCOO in preparation for military operations. The MCOO can cover not just natural terrain but also urban areas.

The MCOO is just one of several tools used by the commander and his staff. During combat operations a Common Operational Picture (COP) is used to maintain a perspective of the friendly and enemy situations. However, this system does not provide the commander with a 3D perspective of the friendly and enemy situation and the terrain. Additionally, the commander and his staff have to depend upon several systems to provide different data sets related to the military operation.

C. MOTIVATION

During my fifteen years of service as a Marine I have had several opportunities to observe how a commander and his staff maintain and manage operational data and a perspective of the battlefield. As a Marine Intelligence officer I have prepared several MCOOs and helped in maintaining a perspective of the enemy and friendly situation. In preparation for a military operation six to twenty four hours can be spent in preparing a picture of the battlefield. The MCOO alone can take several hours to be prepared. Additional time is taken to plot enemy units on the map based upon intelligence data typically taken from a database or new intelligence reports.

One real world operation that I was involved in related to the re-enforcement of an American Embassy. When we were given our warning order my unit was located in the United States. In preparation for our deployment I was tasked to provide the commander with a view of the battle space in order to allow our unit to prepare for combat operations. To do this my section had to first conduct several forms of analysis and then prepare several documents and visual tools. One tool was a MCOO that focused on the Embassy and surrounding urban area. Next we overlaid current defensive positions of units already at the Embassy. We then reviewed video taken from the Embassy and typed up a document to allow the commander to follow along with the video and reference a map of the Embassy and the MCOO. Finally, we had a scale model constructed of the Embassy that we had to carry with us when we deployed. This process took over a week to complete. If a 3D model of the Embassy had existed at the time our task could have been completed within hours. Additionally, the commander could have reviewed defensive positions and ensured coverage of fields of fire. With additional development, a tool could have been integrated with the 3D web-based perspective that would have allowed the commander to run scenarios related our defense plans. The military operations I have participated in and the lessons learned from those operations are my motivation for this thesis work.

1. My Interest and Background

During my time as an intelligence officer I have been involved in several real world combat operations. In all of these operations a great amount of time was wasted in the methods used to prepare a picture of the battlefield. The MCOO is used to help show the commander what effects the terrain will have on operations. Then even more time is spent overlaying the enemy and friendly situation on top of the terrain. These efforts are to help the commander to understand how the terrain might affect combat operations. In all of the combat operations that I was involved computer systems where not used to show the commander a picture of the battlefield. Even with new technologies the military is still spending vast amounts of time in preparing a picture of the battlefield. These systems can show the commander the disposition of friendly and enemy units, but it does not show the commander the terrain.

To truly understand the battlefield the commander must see the battlefield and the operational data related to it in a view that will closely resemble how the battlefield actually appears. Another problem is the ability to bring in different operational data sets that relate to the battlefield. Being able to portray military operations in a intuitive manner and merge different data sets would greatly benefit the warfighter.

2. The Need for 3D Visualization of Data in Military Operations

The exemplar developed in this thesis will demonstrate how we can provide the warfighter with a picture of the battlefield that combines terrain and operational data. The view that is created is displayed in a 3D perspective through a web browser. Because of technological advances in web services, XML, XSLT, databases and Java we can provide a perspective of the battle space that allows us to merge terrain data with operational data. Operational data could be the enemy and friendly situation, data related to the terrain or other intelligence reports from a wide range of collection platforms.

The need for 3D visualization of data in military operations is an assumption and opinion of this author. If we look at America's Army, a game developed by the army as a recruiting tool, we can gain several insights into how 3D visualizations are already affecting combat operations. In an article by Wagner James Au [Wagner 2002], discussions with Army Officers revealed that the game helped in training for combat operations. Training for military operations is only one aspect that shows how 3D visualization can aid in combat operations. In an article by Curtis Blais, Don Brutzman, Doug Horner, and Major Shane Nicklaus, USMC [Blais 2001] they discuss the need for 3D visualization of the battlefield and interest shown by the United Stated Marine Corps. While further research may need to be conducted on the requirements and effectiveness of using a 3D visualization of the battle space in command and control operations, my firm belief is that it will greatly aid commanders in combat operations.

D. OBJECTIVES

It is the objective of this thesis research to demonstrate how military operational data may be represented in a 3D view of the battlefield. It provides an exemplar application that may be used as a model for further development to model other forms of military data related to a battle space. This thesis also provides the reader with an understanding of techniques and procedures needed to develop a 3D battlefield and bring operational data into this 3D view of the battlefield.

E. THESIS ORGANIZATION

Chapter II reviews technologies used in 3D modeling and other research that has been conducted in 3D modeling and modeling of combat operations. Chapter III outlines what commercial tools are used, the technical procedures used to develop terrain and urban environments, and the technical procedures used to integrate operational data from a relational database. Chapter IV discusses the results of the development of 3D models of terrain and urban areas, the performance of these models in a web-based interface, and the results of merging operational data from a relational database in a 3D battlefield. Chapter V covers recommendations, conclusions, technical findings, and summarizes the conclusions and recommendations for future work.

II. RELATED WORK

A. INTRODUCTION

The following topics will be discussed in this section:

XML – An extensible mark-up language that uses tags like HTML but allows the user to define the tags.

XSLT – An extensible language that uses tags like XML that is used to transform XML document structure.

JAVA – Programming language developed by Sun Microsystems.

Extensible 3D (X3D) Graphics – A 3D graphic modeling language.

Major Shane Nicklaus' Thesis: XML OPORDER VISUALIZATION – A thesis that discusses the use of XML to enable the visualization of operation orders.

LT James Harney's Thesis: AGENT-BASED ANTI-TERRORISM/FORCE PROTECTION (AT/FP) MODEL – A thesis that discusses the development of an anti-terrorism and force protection model for use by the United States Navy.

B. RELEVANT WEB TECHNOLOGIES

1. XML

XML is a subset of the Standard Generalized Markup Language (SGML). [Hunter 2001] XML is not a true computer language. Like HTML, XML is comprised of tags. These tags are words that are found within brackets (<…>). Unlike HTML the words found between the brackets are determined by the user. The meaning of the tags in HTML is predetermined to allow a web browser to interpret the tags and display the information in a web browser. XML tags are undefined. This allows the user to describe data within the tags in his own language. Figure 3 is an example of an XML document. XML is not completely without structure. There are rules that must be followed to allow applications to be able to read the XML documents. Chapter two of the Hunter 2001 reference provides a good description of the rules needed to make an XML file well-formed and valid.

```
<?xml version=”1.0" encoding="UTF-8"?>
<Target>
   <targetId>001</targetId>
   <targetName>Baghdad Military HQ</targetName>

</Target>
```

Figure 3. Simple example of an XML document.

The important thing to take away from XML is that it allows you to describe data in your own language and check its validity. Because of the structure of XML it can be read by the user, easily understood without the use of a computer application and can be understood by software applications. Because XML has a universal format it allows programmers to pass data from one application to another. This flexibility is what makes XML a very powerful tool in programming and in the exchange of data.

2. XSLT

Extensible Stylesheet Language for Transformation (XSLT) is a language for transforming the structure of an XML document. [Kay 2001] The exemplar in this thesis demonstrates how XSLT can take an XML document and transform the data within that document into X3D. By using another XSLT users or programmers can transform the X3D document into VRML. XSLT is an enabling technology that helps to make XML a powerful tool in programming and in the exchange of data.

3. JAVA

Java is a programming language developed by Sun Microsystems. Java is a Web-friendly language that is cross-platform capable at run-time without having to recompile. [Harney 2003] Java is used in this thesis to facilitate the application of the two XSLT documents used. For more information on Java please visit http://java.sun.com/ (accessed August 2003).

4. Extensible 3D (X3D) Graphics

X3D is an example of the power of XML. X3D is a developing 3D graphics construct that is the follow-on to the VRML modeling language. Because X3D is XML it is the enabling technology used to allow relational data to be viewed with a 3D scene. For more information on X3D please visit http://www.web3d.org/x3d.html (accessed August 2003).

C. RELATED RESEARCH

1. Major Shane Nicklaus' Thesis: XML OPORDER VISUALIZATION

This thesis was a great introduction into the capabilities that a 3D battlespace can provide to today's warfighters. The thesis was based upon modeling an amphibious operation using X3D and XML. Concepts in Nicklaus' thesis served as a starting point for this thesis. For more detail please refer to the reference for this thesis. [Nicklaus 2001]

2. LT James Harney's Thesis: AGENT-BASED ANTI-TERRORISM/FORCE PROTECTION (AT/FP) MODEL

Harney's thesis demonstrated that X3D and XML can be used to develop tools that allow military planners to test defenses against possible terrorist attacks. An exemplar was developed that allows the user to test defenses against a possible surface attack against a naval vessel in harbor. The exemplar used X3D, XML and JAVA technologies. For more detail please refer to the reference for this thesis. [Harney 2003]

D. SUMMARY

This section has covered technologies and related research pertinent to this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DEVELOPMENT OF 3D SCENES, DATABASE AND EXEMPLAR

A. INTRODUCTION

The technology and methods used to develop 3D models are not new concepts. However, the technical approach and methods to modeling of operational data in a web-based 3D model is a new area of research. Research outlined in this chapter can serve as a guide to the warfighter who desires to develop a web-based 3D perspective of the battlefield.

This chapter walks through the development of several models that provide a web-based 3D perspective of a battlefield. The chapter is organized based upon the approach taken by this author during the course of the research in modeling operational data. Several tools are used during the process of developing models. A brief description is provided of each tool. The processes and functionality used by a particular application in developing a 3D model of the battlespace are also covered. This chapter is not intended to provide a manual for these applications nor be a promotion or evaluation of the tools.

B. DEVELOPMENT TOOL DESCRIPTION

1. X3D-Edit

X3D-Edit is a graphics file editor for Extensible 3D (X3D) Graphics. It is used in authoring and programming of X3D scene-graph files. The simplicity of the tool and the associated help file promote error free editing. This tool is critical in development of the 3D battlespaces used in this thesis. It helps to ensure compliance with the X3D and VRML 2.0 standards. The X3D files serve as a basis for conversion of operational data from a relational database into the VRML scene-graph files. For a more detailed description of the tool please visit http://www.web3d.org/TaskGroups/x3d/translation/README.X3D-Edit.html (accessed August 2003).

2. VRMLPad 2.0

VRMLPad 2.0 is a VRML scene-graph editing tool developed by Parallel Graphics. It allows quick and easy editing of VRML scene-graph files. It is very helpful in displaying large scenes, ensuring VRML compliance, and error free editing. It also displays the routing used to control movement within a VRML model. For a more detailed description of the tool please visit http://www.parallelgraphics.com/products/vrmlpad (accessed August 2003).

3. Microsoft's WordPad

WordPad is a text editor. It allows several different file formats to be opened. While not a X3D or VRML editor it can be used to open and edit both X3D and VRML files. It does not help in ensuring error free editing or compliance to the VRML specification. For a more detailed description of the tool please visit http://www.Microsoft.com (accessed August 2003).

4. XML SPY

XML SPY is an XML editor developed by Altova. This tool provides a robust functionality when working with XML files. The tool also allows editing and debugging of XSLT files. It also checks XML against a Document Type Definition (DTD). For a more detailed description of the tool please visit http://www.xmlspy.com (accessed August 2003).

5. Stylus Studio

Stylus Studio is an XML editor developed by Sonic Software. During the research conduct in this thesis Stylus Studio is predominantly used in the development of XSLT files. The tool is extremely helpful in developing and modifying the XSLT code. For a more detailed description of the tool please visit http://www.sonicsoftware.com/products/additional_software/stylus_studio/index.ssp (accessed August 2003).

6. Polytrans for Windows

Polytrans for Windows was developed by Okino Computer Graphics. This program is used to translate models from OpenFlight format to VRML 2.0. The tool supports translation of several file formats. For a more detailed description of the tool please visit http://www.okino.com/conv/conv.htm (accessed August 2003).

7. Internet Model Optimizer

The Internet Model Optimizer is provided by Parallel Graphics. This tool is used to help reduce the number of polygons found in the more complex 3D models and the terrain developed. When working with models that were not traditionally built to be viewed through a web browser this tool proves to be extremely useful. An additional benefit of this tool is its ability to allow color changes to be made to models that do not use textures. For a more detailed description of the tool please visit http://www.parallelgraphics.com/products/imo (accessed August 2003).

8. MultiGen Creator V2.5

MultiGen Creator is provided by MultiGen-Paradigm. This tool is a software application that allows for development of highly detailed models and terrain. All terrain used in this thesis is built using the application. The built-in ability of the application to read and convert DTED data is extremely useful. The terrain building application within the tool allows for the output of several different terrain map projections and Earth Ellipsoid Model. The most commonly used map projections in this thesis are Flat Earth and Universal Transverse Mercator (UTM). For a more detailed description of the tool please visit http://www.multigen.com (accessed August 2003).

9. Microsoft's SQl Server

The database used for this project is Microsoft's SQL server. While this prototype uses Microsoft's SQL server the actual concepts involved can be reconstructed on most major database servers to include a native XML server. The main trait that is needed is the ability of the server to output data in an XML format. An update titled SQLXML had to be added to allow for the output of XML files. For a more detailed description of the tool please visit http://www.microsoft.com/sql (accessed August 2003).

10. Tomcat Server

The Tomcat server is an open source tool distributed by the Jakarta Project. The Tomcat server is used to run the Java servlets used in this project. For a more detailed description of the tool please visit http://jakarta.apache.org/tomcat (accessed August 2003).

11. VRML Devicehandler by Geovrml.com

This application was developed by geovrml.com. The application allows for the use of an external device to navigate and control objects within a VRML scene. This application allows for the programming of an aircraft to enable the user to fly around in the scene with little effort or knowledge of the controls. For a more detailed description of the tool please visit http://www.geovrml.com (accessed August 2003).

12. Microsoft's Internet Information Services (IIS)

This is a web server developed by Microsoft. This web server is used in this thesis for all web pages. For a more detailed desc40ription of the tool please visit http://www.microsoft.com/WindowsServer2003/iis (accessed August 2003).

13. Internet Explorer

This is a web browser developed by Microsoft. This is the web server used in this thesis project for all 3D models viewed with the Cortona plug-in for web browsers. For a more detailed description of the tool please visit http://www.microsoft.com/windows/ie/default.asp (accessed August 2003).

14. Cortona VRML Client 4.1

This is a Plug-in for web browsers that is used to view 3D models developed in VRML. Cortona is distributed by Parallel Graphics. This VRML viewer is used to view all 3D models developed for this thesis project. For a more detailed description of the tool please visit http://www.parallelgraphics.com/products/cortona (accessed August 2003).

C. DEVELOPMENT OF A 3D MODEL FOR URBAN OPERATIONS

1. Overview

Through the course of research two models are developed to represent an urban area. These two models range in scale from about two dozen buildings to several hundred buildings. Several commercial applications are used to develop these models. The first model is a representation of the NPS campus. The second is a model that covers a large portion of the city of Baghdad.

2. Naval Postgraduate School Campus Model

a. The Old Campus Model

The Naval Postgraduate School campus model was originally built by John Locke, a research associate at NPS. The model was built in a Multigen Creator in a OpenFlight file format. The original model was built from a line drawing of the campus (Figure 4). The model was converted into an X3D and VRML format by Jeff Weekley and Don Brutzman in April 2002 (figure 5). The original model is an excellent representation of the campus (Figure 6, 7).

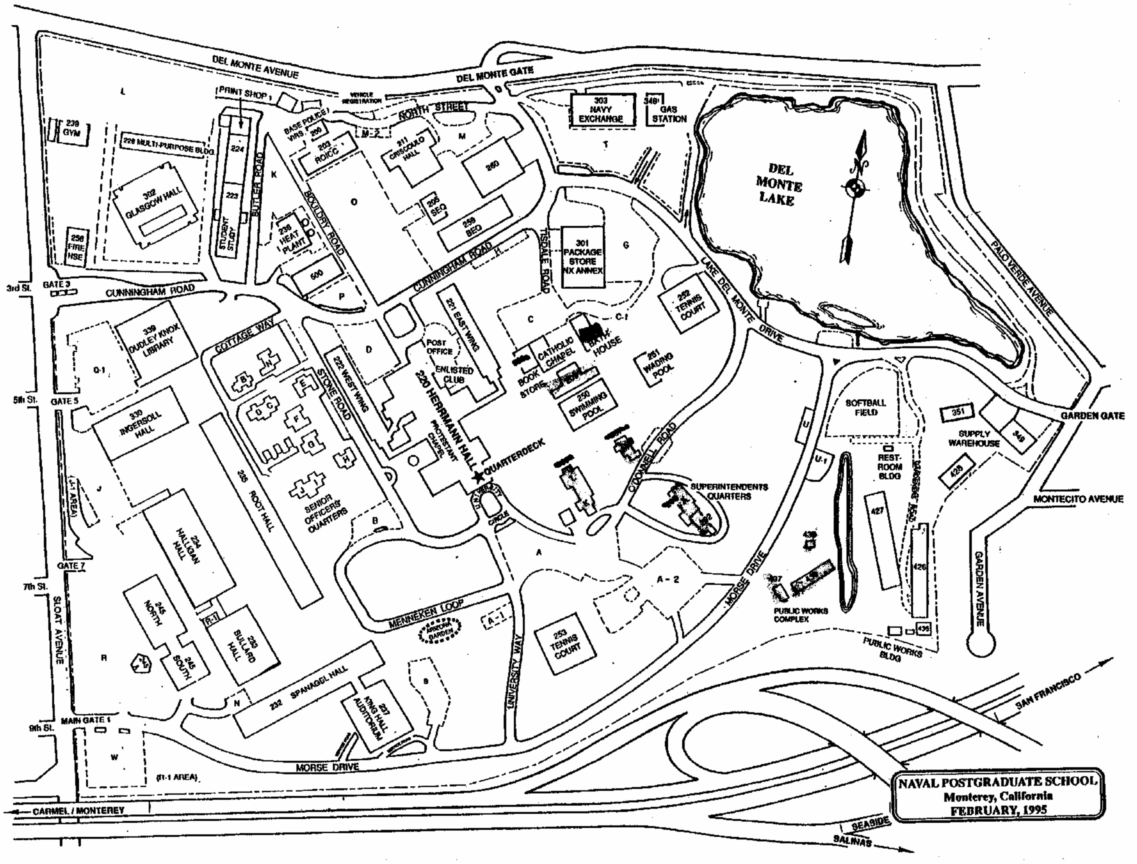


Figure 4. Map of NPS campus published by the school in February, 1995.

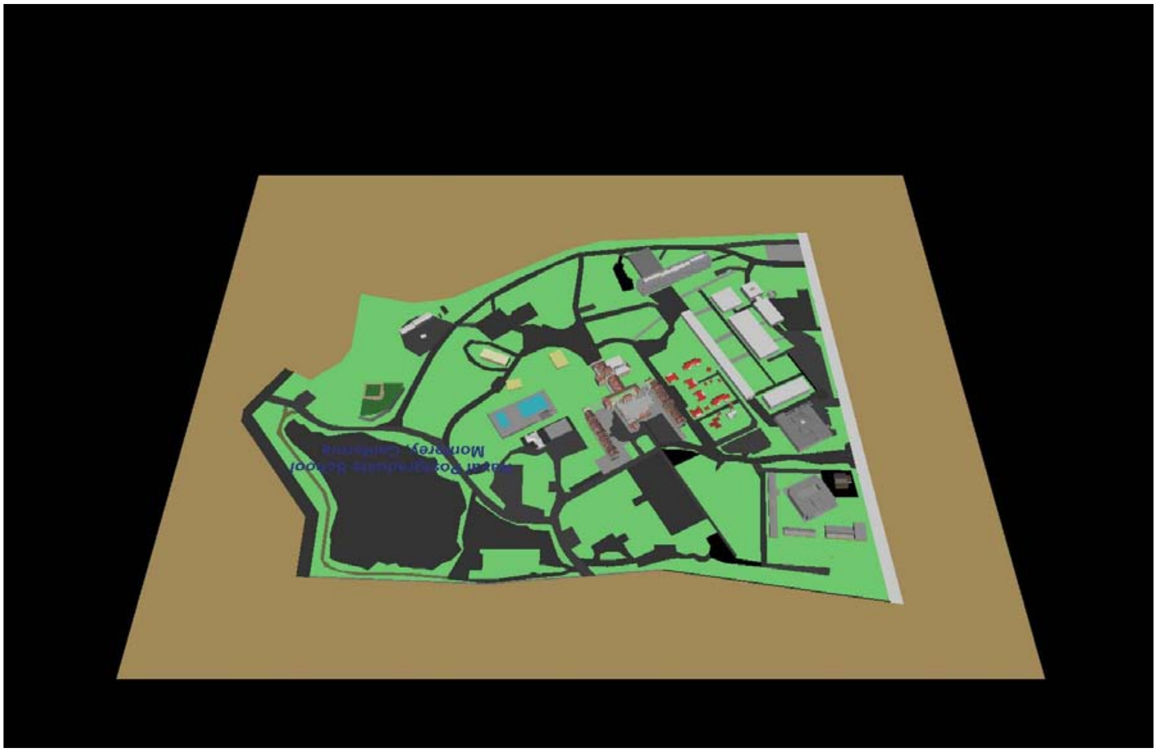


Figure 5. Screen shot of the original 3D campus model in X3D.

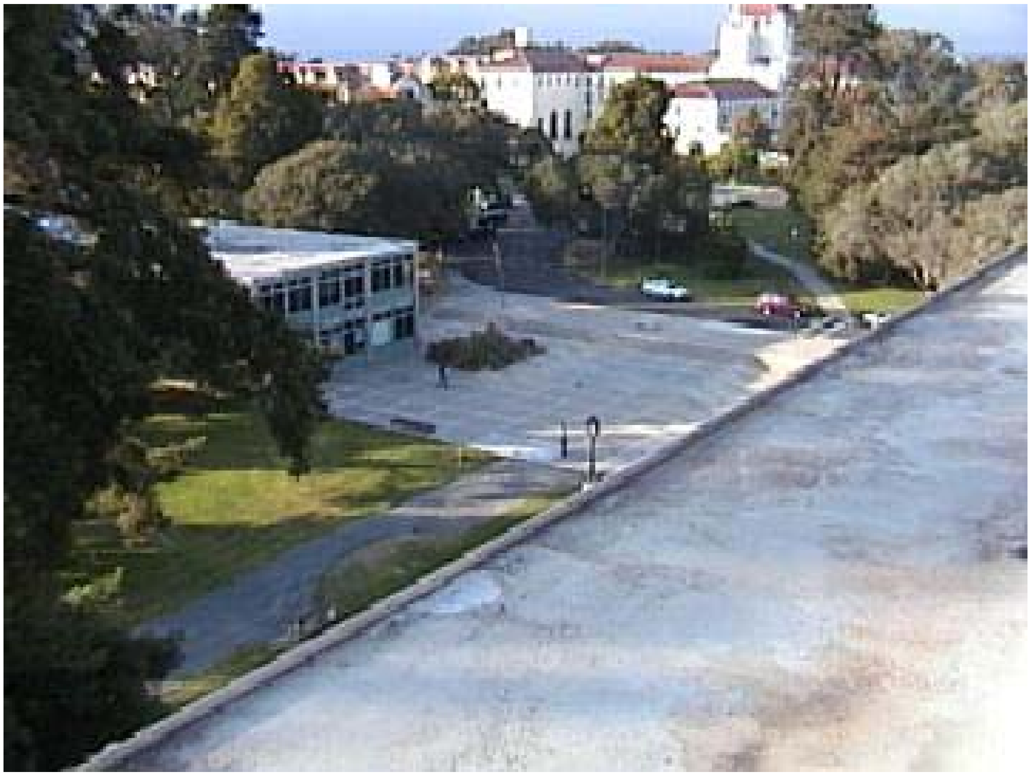
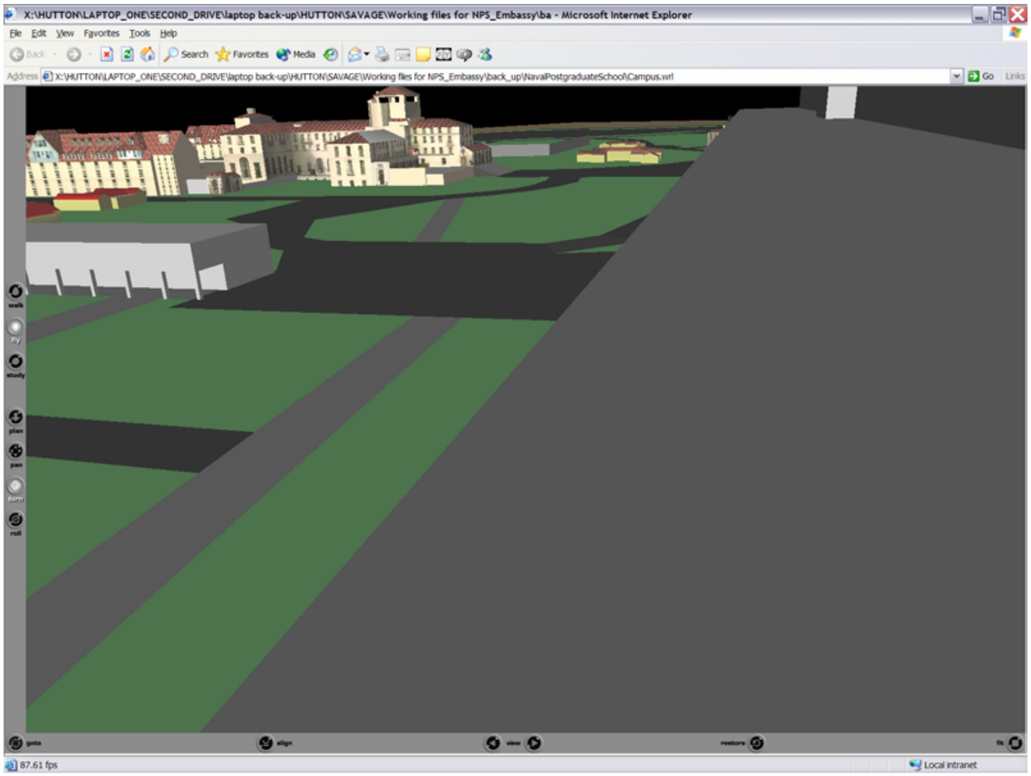


Figure 6. Picture taken from top of Spanagel Hall looking toward Root Hall on July 28, 2003.



X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\SAVAGE\Working files for NPS_Embassy\ba - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\SAVAGE\Working files for NPS_Embassy\back_up\NavalPostgraduateSchool\Campus.wrl
Go Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
87.61 fps
Local intranet

Figure 7. Old NPS Campus model showing same position as photo in Figure 6.

b. Reasons to Build a New Model

Closer examination of the original campus model revealed a few discrepancies that brought about the need to develop a newer version of the campus model. The scaling and location of buildings in the model was not precise. Comparing the campus map with the original campus model shows the difference in scaling and location of buildings (Figure 8, Figure 9). Additionally, the model was built before some major construction had occurred on the campus. In order to update the scaling and add and remove several buildings, a new model was needed.

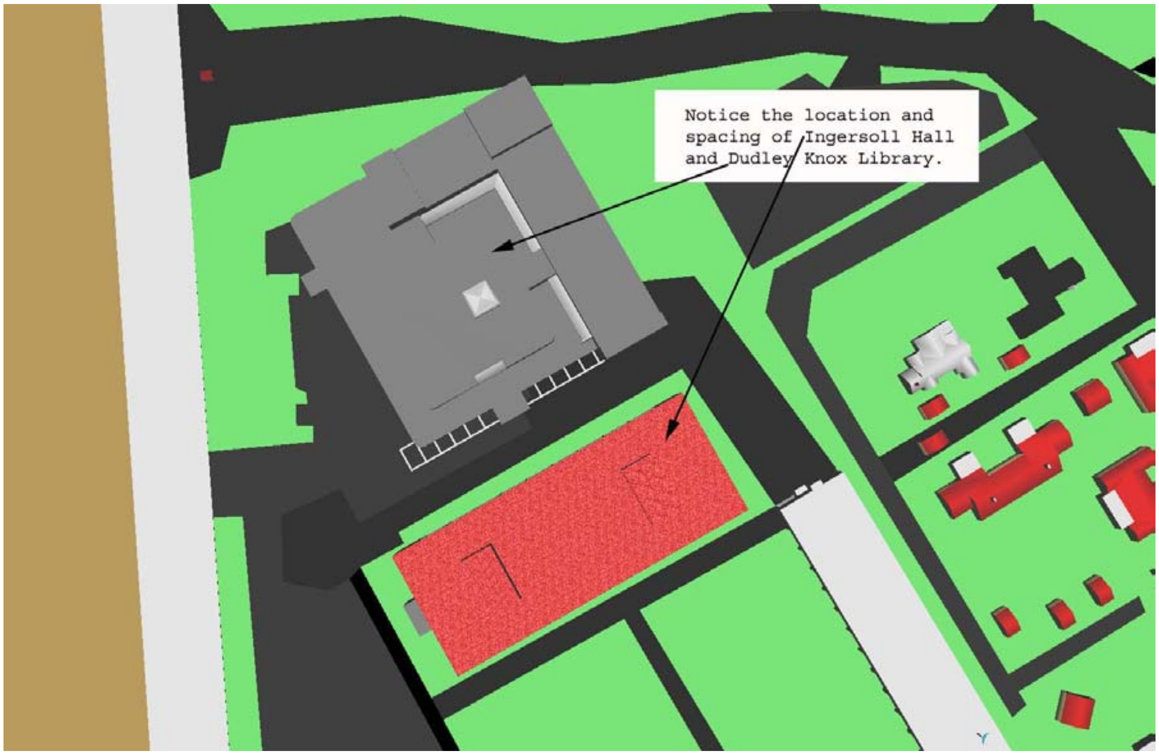


Figure 8. Screen shot of old campus model showing the spacing and location of two buildings.

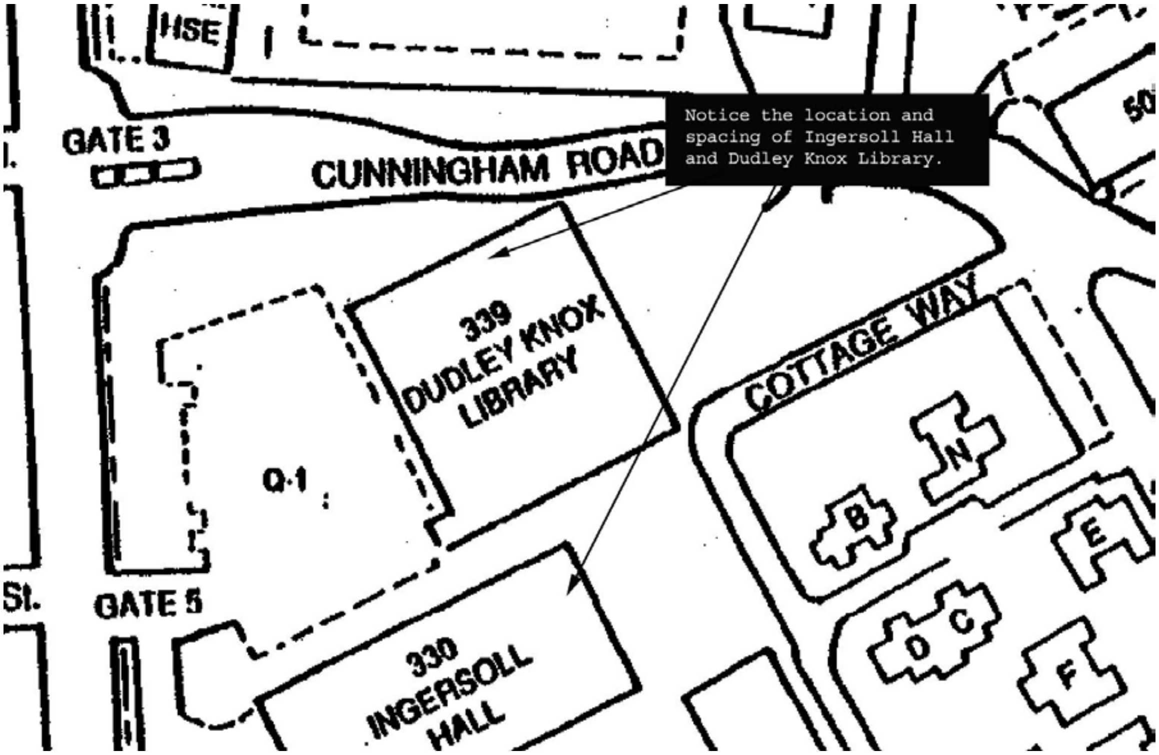


Figure 9. Campus map showing the spacing and location of the same two buildings shown in Figure 8.

c. Establishing the Foundation/Terrain

In order to ensure the scaling and location of the buildings in the new model of the campus, an aerial photo of the campus was taken. The aerial photo was taken from a Cessna airplane piloted by a fellow student (Figure 10). The process of taking the photo took two attempts. In the first attempt, the photo was taken from an altitude that was not high enough to include the entire campus. Because of the difference in location of the photos taken it proved to be impossible to combine the photos using Adobe Photoshop. The second attempt at higher altitude allowed the entire campus to be captured in the photograph. Another factor that was important to the photo was to try and take a photo that was almost directly over the campus. This was important to ensure the photo would be more realistic when used in the model.

Figure 10. Airplane used to take aerial photo of campus.

The aerial photo taken of the campus was used as a texture placed on top of an elevation grid (Figure 11). Before the photo was used in the 3D model several modifications were made to the photo. The photo was cropped to only show the area of coverage needed. A grid system was added to help with placing buildings and objects. The last thing done to the photo was to flip the photo horizontally. This was done because the photo shows up reversed when displayed on the elevation grid (Figure 12).

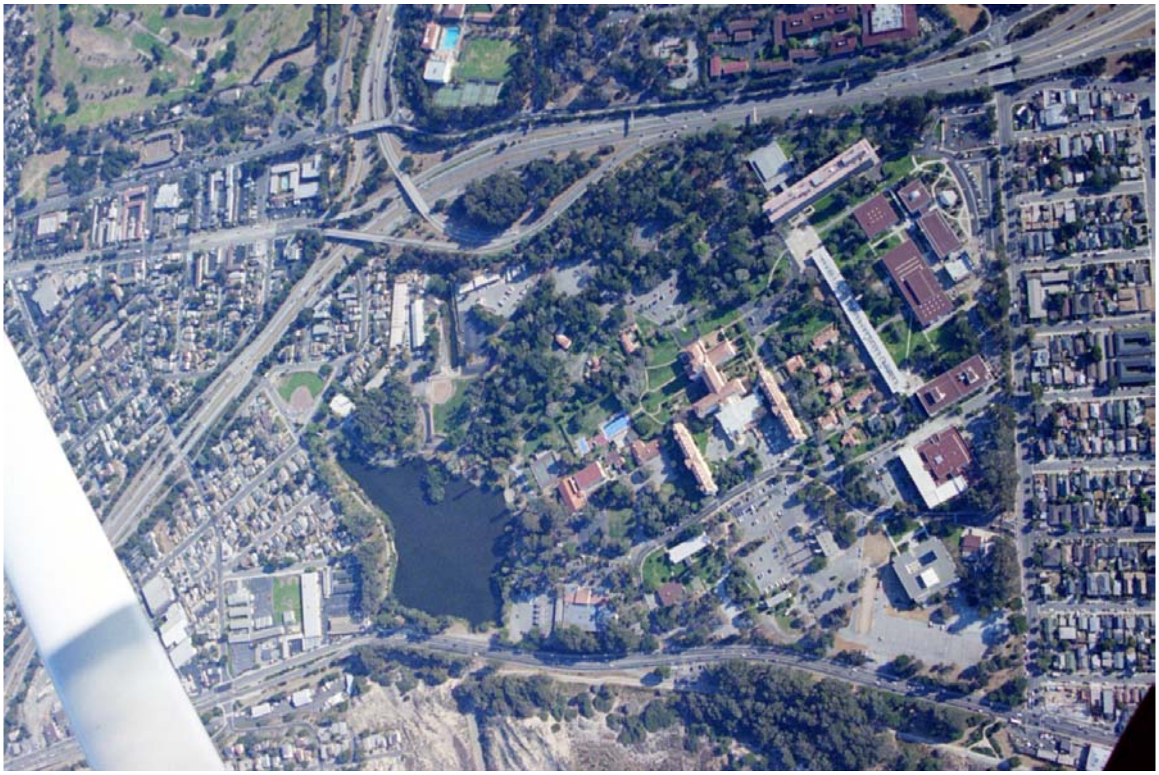


Figure 11. Original photo taken of campus in the year 2002.

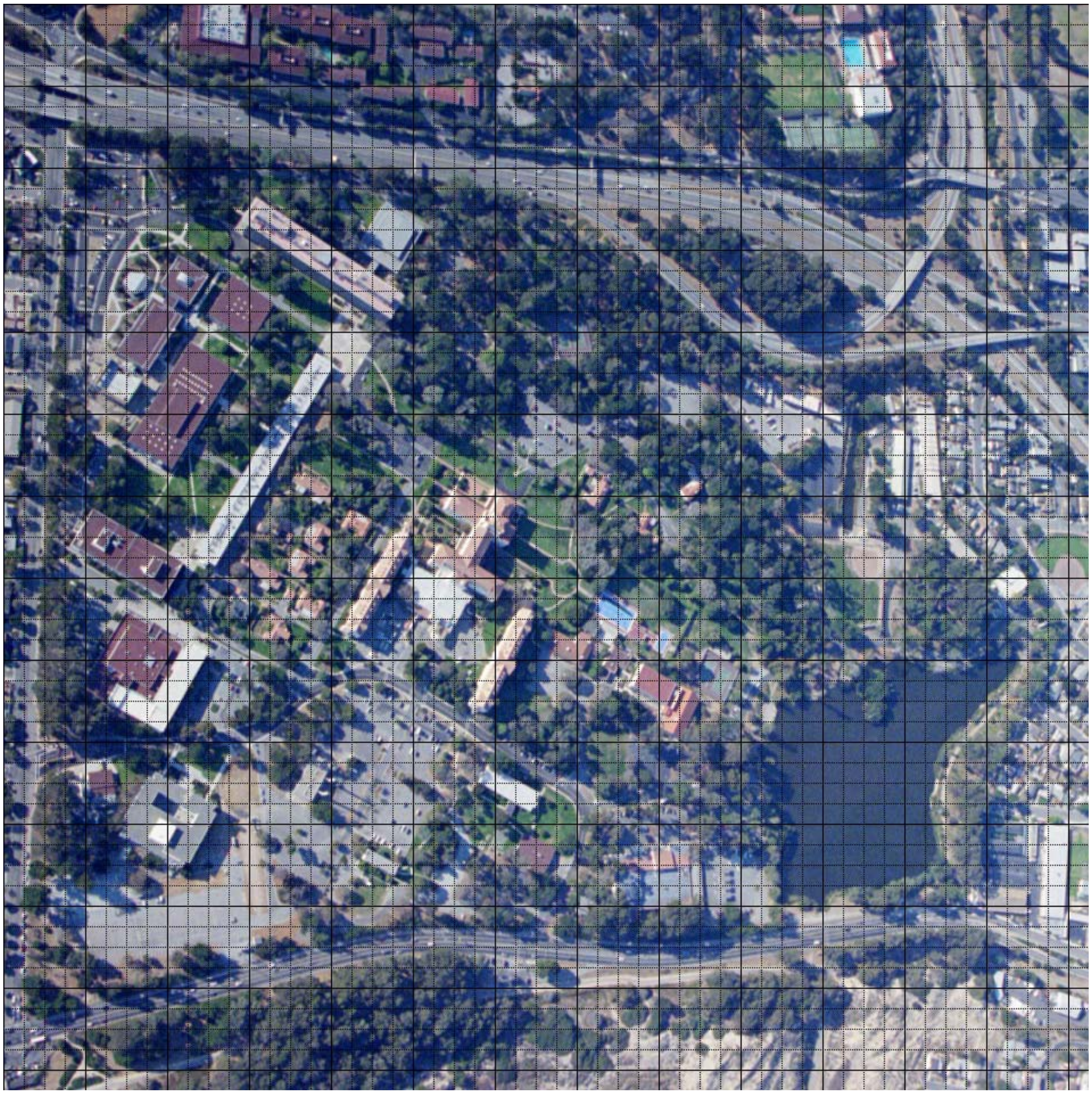


Figure 12. Image of campus after modifications to the photo in Figure 11.

After the image was modified, it was placed on an elevation grid in X3D. Using an elevation grid allows for future modification. The current elevation grid has zero for all elevation points in the grid system. If elevation data is ever obtained for the campus then the elevation grid can be filled in to reflect the true elevation found on the campus.

d. Placing Buildings and Resizing

The buildings in the old model were reused in the new model. Some buildings were not used from the old model because these buildings had been destroyed and no longer exist on campus. The grid system placed on the final image makes the process of placing objects relatively simple. To place an object in X3D an X, Y, Z system is used (Figure 13). All objects placed in the seen are placed in the positive X-Z plane. The Y location is slightly adjusted for each building. Scaling and the exact position of each building are accomplished by using the campus image placed upon the elevation grid. The campus image ensures that the scale and location of the buildings are precise. Nearly every building required rescaling for the new model.

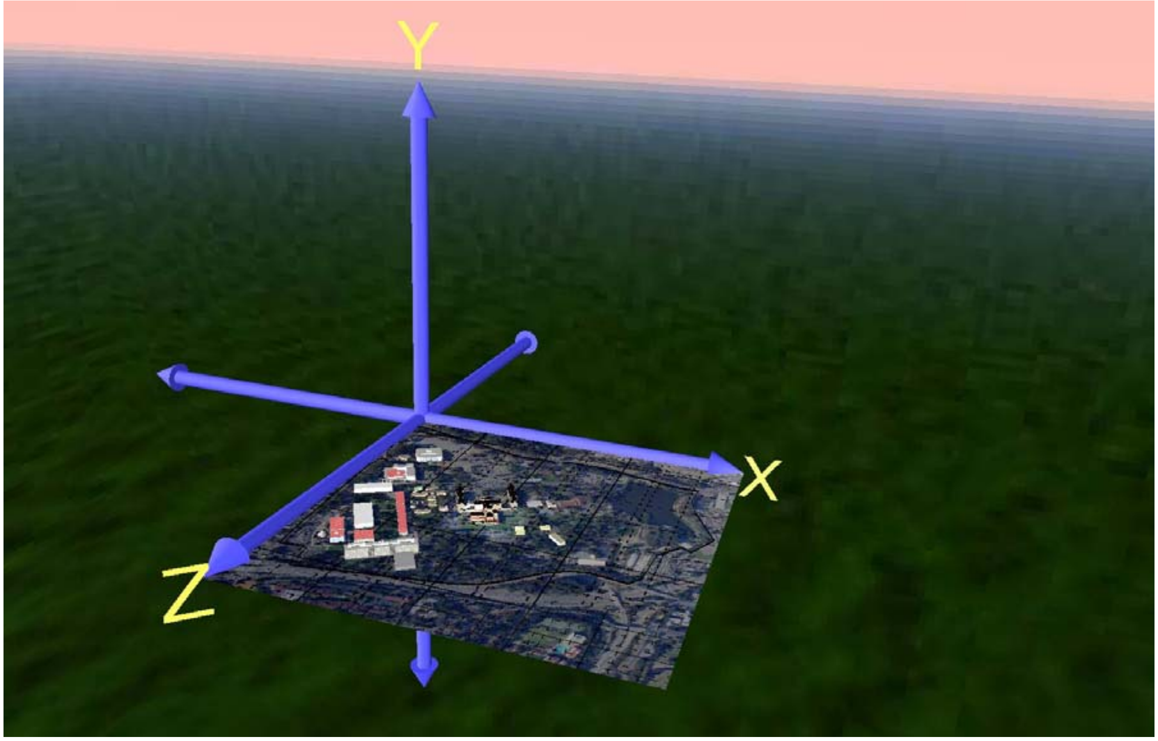


Figure 13. NPS Campus model with an X, Y, Z axes.

e. Adding a Fence

One model that had to be rebuilt was the fence that goes around the campus. The fence in the old model could not be scaled and was not correct in relation to the actual location. The new fence was constructed by John Locke by using the aerial photo in MultiGen Creator to ensure the exact location of the fence. The model was then converted the model into VRML using Polytrans. After the model was converted it was then added to the new campus model.

f. *Adding a New Herrmann Hall*

One other change made to the campus is a high resolution model of Herrmann Hall developed by John Locke. This model is very different from the old model of Herrmann Hall. The model has all major rooms inside of the building developed (Figure 14).

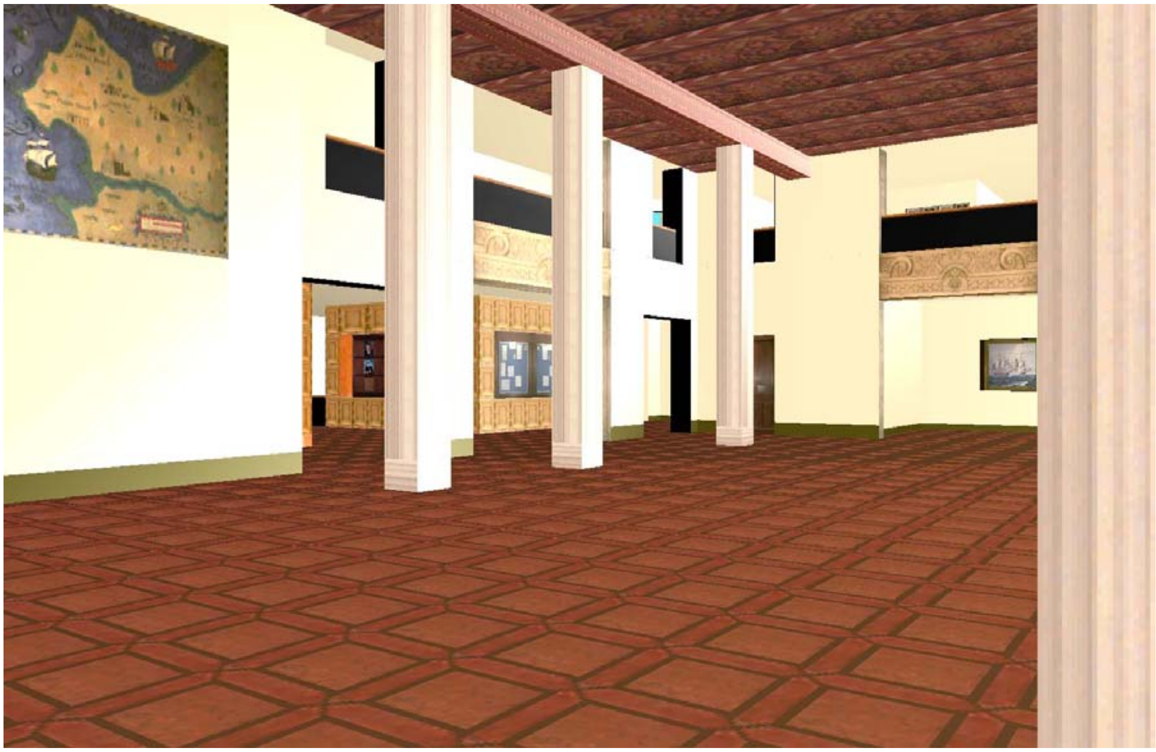


Figure 14. Inside screen capture of the Herrmann Hall Model.

g. *Resizing of Textures to a Power of Two*

To improve performance in rendering of the new campus model all images used as textures in the model were modified. When images are used as textures within a 3D model the size of the image is very important. If the image is not a power of two then the graphics card is required to perform more calculations in order to render the image. All browsers read any texture in any size the user provides, but when rendered, they may be scaled to fit the requirements of whatever rendering engine is used for that browser. [Roehl 1997] When the image is sized to a power of two then fewer calculations are required and performance in rendering the model is increased. Since the new model changed dramatically from the old model no measurements were made to compare performance. An estimated 162 images were modified using Adobe Photo Shop.

h. Current Status of Model

The new model of the campus reflects a version of the campus as it would have been in October of 2002 (Figure 15). Since that time at least one new building has been added to the campus. The current model and the old model did not have all buildings that are located on the campus. Currently the new model can be viewed at

http://web.nps.navy.mil/~brutzman/Savage/Locations/NavalPostgraduateSchool/chapter.html (accessed August 2003).

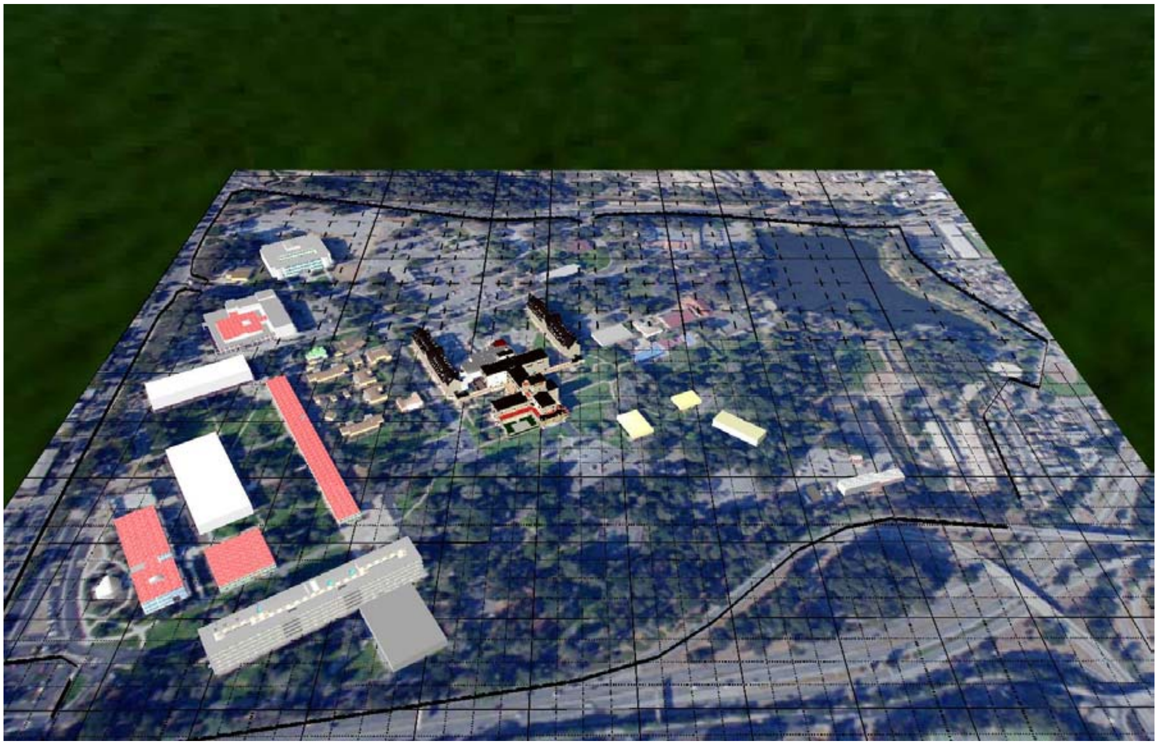


Figure 15. Screen shot of new NPS campus model with 3D buildings placed on top of a geo-referenced image-map photograph.

3. Baghdad Model

a. The Original Model

The original Baghdad city model was built by the contractor Object Raku through the US ARMY Research, Development and Engineering Command, Simulation Technology Center, 12423 Research Pkwy, Orlando, FL. The model does not cover the entire city. The original model comes with four versions. Two versions are in an Open

Flight format. The Open Flight formats include a low-resolution and a high resolution version. The other two formats are in VRML formats. One of the VRML formats includes specific coding that required a Contact VRML plug-in from Blaxxun. The second version is a normal VRML model (Figure 16).

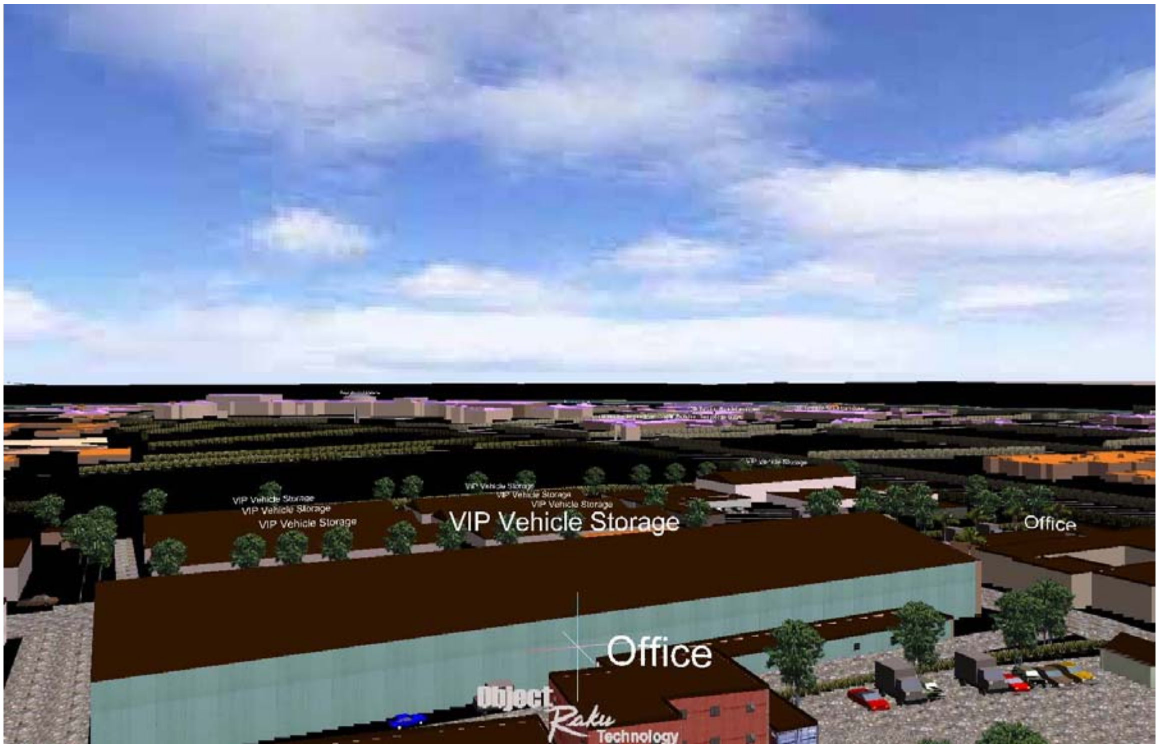
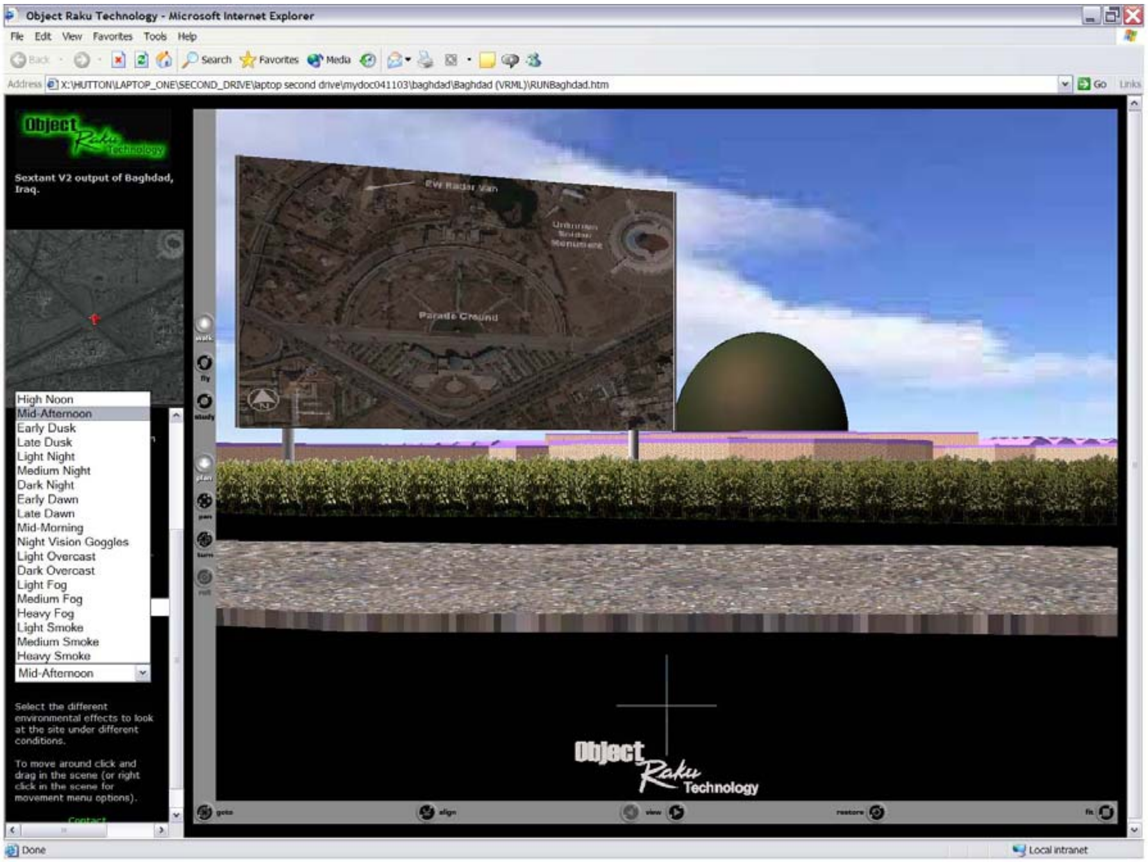


Figure 16. Screen shot of Baghdad VRML model by the contractor Object Raku through the US ARMY Research, Development and Engineering Command, Simulation Technology Center.

b. Conversion of Model from OpenFlight Format Versus the VRML Format

To use this model for further research a decision had to be made regarding which version to use. The end result needed is to be a model in X3D. Upon inspection of the code for both of the VRML models, it was determined the easier task is to convert the OpenFlight format models into VRML then X3D. This is because the developers tied the entire scene into switch statements that change the appearance of the scene based upon the time of day chosen by the user (Figure 17).



Address: X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop second drive\mydoc041103\baghdad\Baghdad (VRML)\RUNBaghdad.htm

Sextant V2 output of Baghdad, Iraq.

Mid-Afternoon

Select the different environmental effects to look at the site under different conditions.

To move around click and drag in the scene (or right click in the scene for movement menu options).

Figure 17. Baghdad model showing popup selection for different times of day and weather conditions.

c. *Conversion Process*

The conversion of this model is still ongoing. One issue with the conversion of this model is the size of the model. The low-resolution Open Flight format is over 116 megabytes in size. Due to the size of the model and limitations of the laptop used assistance was obtained from Matt Prichard, a research associate at NPS. While in the MultiGen Creator tool items can be selected and grouped in an order desired by the user (Figure 18). The model was broken out into different sections. These sections include items like trees, ground, buildings, roads, and cars. After deleting sections like trees, terrain, and cars, the model was then converted into VRML.

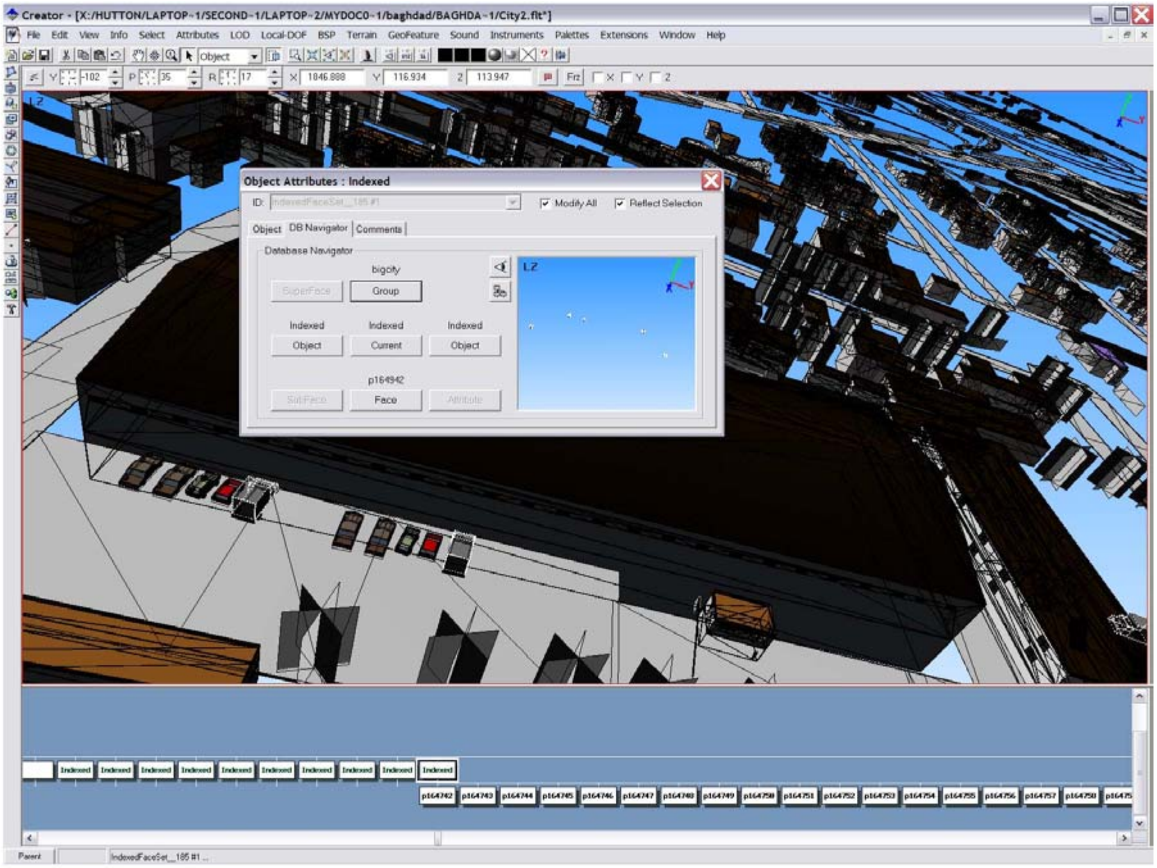

Creator - [X:/HUTTON/LAPTOP~1/SECOND~1/LAPTOP~2/MYDOC0~1/baghdad/BAGHDA~1/City2.flt*]
File Edit View Info Select Attributes LOD Local-DOF BSP Terrain GeoFeature Sound Instruments Palettes Extensions Window Help
Object
Y -102 P 35 R 17 X 1846.888 Y 116.934 Z 113.947
Frz X Y Z
Object Attributes : Indexed
ID: IndexedFaceSet__185 #1
☑ Modify All
☑ Reflect Selection
Object
DB Navigator
Comments
Database Navigator
bigcity
SuperFace
Group
Indexed
Object
Indexed
Current
Indexed
Object
p164942
SubFace
Face
Attribute
Indexed
p164742 p164743 p164744 p164745 p164746 p164747 p164748 p164749 p164750 p164751 p164752 p164753 p164754 p164755 p164756 p164757 p164758
Parent
IndexedFaceSet__185 #1 ...

Figure 18. Screen shot from MultiGen Creator tool showing item grouping.

d. Current Status of Model

The model is partially in a VRML format. The current state of the model is usable for future studies. Work needs to be conducted on the model in order to lower the frame rate when viewing the scene. The new VRML model is just over 30 megabytes. Future development is recommended with this model (Figure 19).

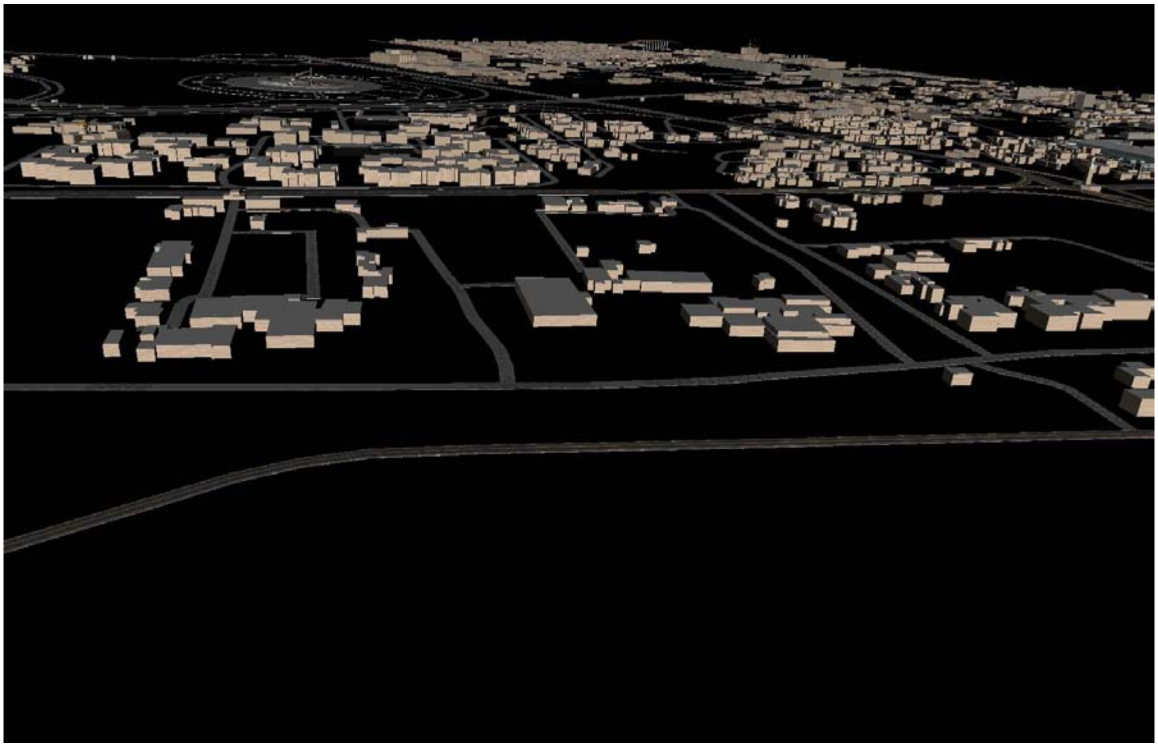


Figure 19. Screen shot of new Baghdad VRML model currently being developed.

4. Recommendations and Conclusions

a. Overview

The development of the campus model and the Baghdad model took several months to accomplish. It is important to remember that both of the models can be viewed through a web browser.

b. Recommendations for Further Development of the Campus Model

The foundation for this model is now well established. To add to this model it is first recommended that the rest of the buildings on campus be added. Second the tree growth that is found on campus can be added. Finally, elevation data can be added when it is obtained.

c. Recommendations for Further Development of the Baghdad Model

This model still requires extensive work. The size of this model still needs to be reduced. This can be accomplished by first replacing the high polygon count buildings with lower polygon count representations. Another step to take is to use the X3D Geospatial Profile node called Inline. This node allows for objects in an inline node to be dynamically loaded and unloaded from memory by the use of a proximity sensor. One additional feature is to add a city map texture and map to an X3D Geospatial Profile model.

d. Conclusions

While a very time-consuming task to develop the two models, they provide a very good representation of urban areas. It is apparent from development of these models that several applications are needed in the development of such complex models. Both of these models prove that development of urban models for combat operations is feasible and can be pursued in the future for military applications.

D. DEVELOPMENT OF A 3D MODEL FOR TERRAIN WITHIN THE AREA OF OPERATION (AO)

1. Overview

Two models are developed to demonstrate the ability to model a large area of operation. The first model is of the island of Oahu. The second model is the country of Iraq. Both models use DTED data and map textures for terrain. Both models were completely developed during the course of this research.

2. Island of Oahu

a. Obtaining the DTED Data

There are several levels of DTED data. The levels of DTED data represent the spacing of the elevation postings. The Oahu model uses level two DTED which provides a posting of elevation every thirty meters (approximately). Obtaining the DTED data for Oahu was easy since all the data was on one CD from the National Imagery and Mapping Association (NIMA). MultiGen Creator reads DTED data. After inserting the DTED CD into the computer the user, opens MultiGen Creator and select read DTED CD (Figure 20).

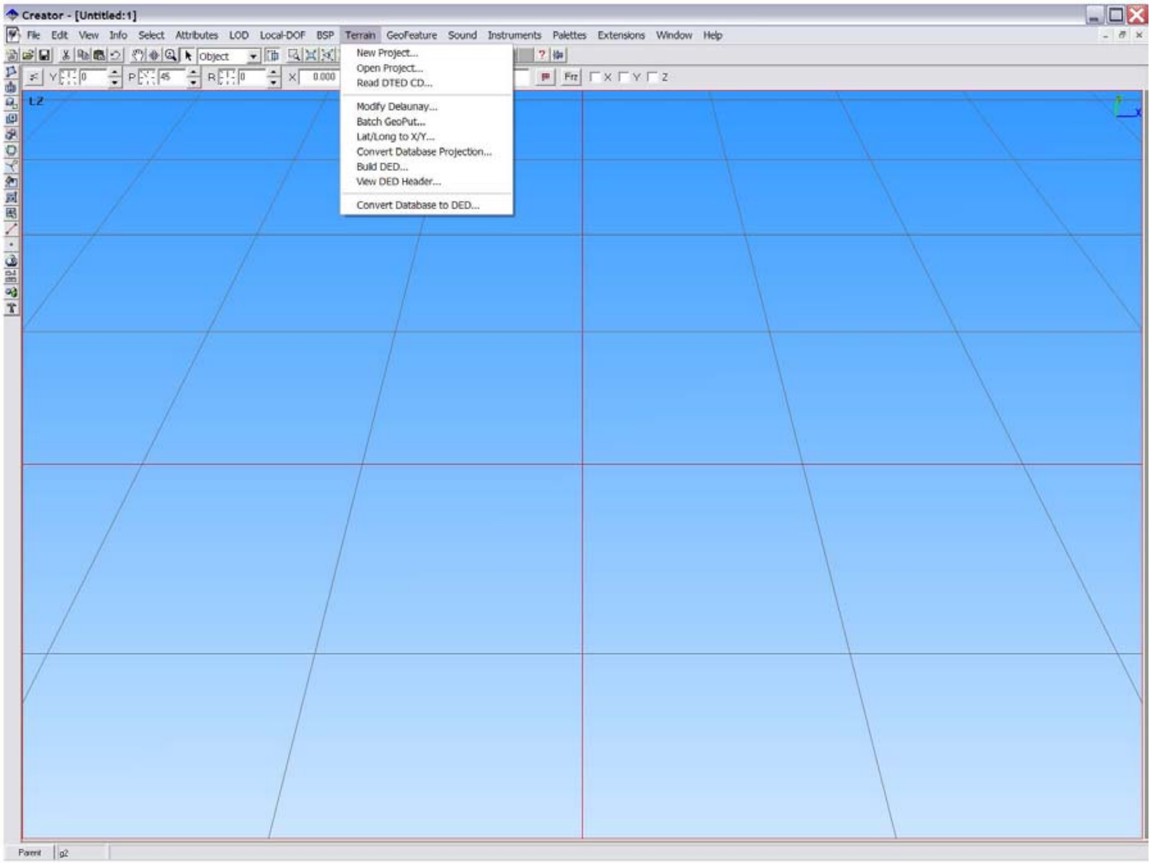

Creator - [Untitled:1]
File Edit View Info Select Attributes LOD Local-DOF BSP Terrain GeoFeature Sound Instruments Palettes Extensions Window Help
New Project...
Open Project...
Read DTED CD...
Modify Delaunay...
Batch GeoPut...
Lat/Long to X/Y...
Convert Database Projection...
Build DED...
View DED Header...
Convert Database to DED...
Parent
g2

Figure 20. MultiGen Creator screen shot showing ability to read DTED data.

b. Elevation Grids Versus Indexed Face Set Terrain

There are several methods for developing terrain within a 3D model. One of the first choices to be made is between using an elevation grid system or an indexed face set system. Regardless of the method used to build the model the rendering of the elevation grid and indexed face set can appear the same within a 3D scene. The major difference is the area of coverage and level of detail rendered by each system. Elevation grids can provide for a greater level of detail at a loss of area of coverage. The exact area of coverage is related to the amount of memory in the computer system used.

Indexed face sets allow for a greater area of coverage but may sacrifice the level of detail depending upon the method used to build the indexed face sets. In order to cover a larger area indexed, face sets are used to generate the Oahu terrain.

c. Mesh Versus Delaunay Terrain

Two methods used by MultiGen Creator to develop indexed face sets are Delaunay and polymesh. Terrain created using the Delaunay method generates irregular shaped polygons (Figure 21). If the polymesh (Figure 22) method is used,uniform polygons are generated that are similar to the polygons generated by using an elevation grid (Figure 23). The Delaunay method in MultiGen Creator is used in this project. This provides a larger area of coverage to be built in a single file.

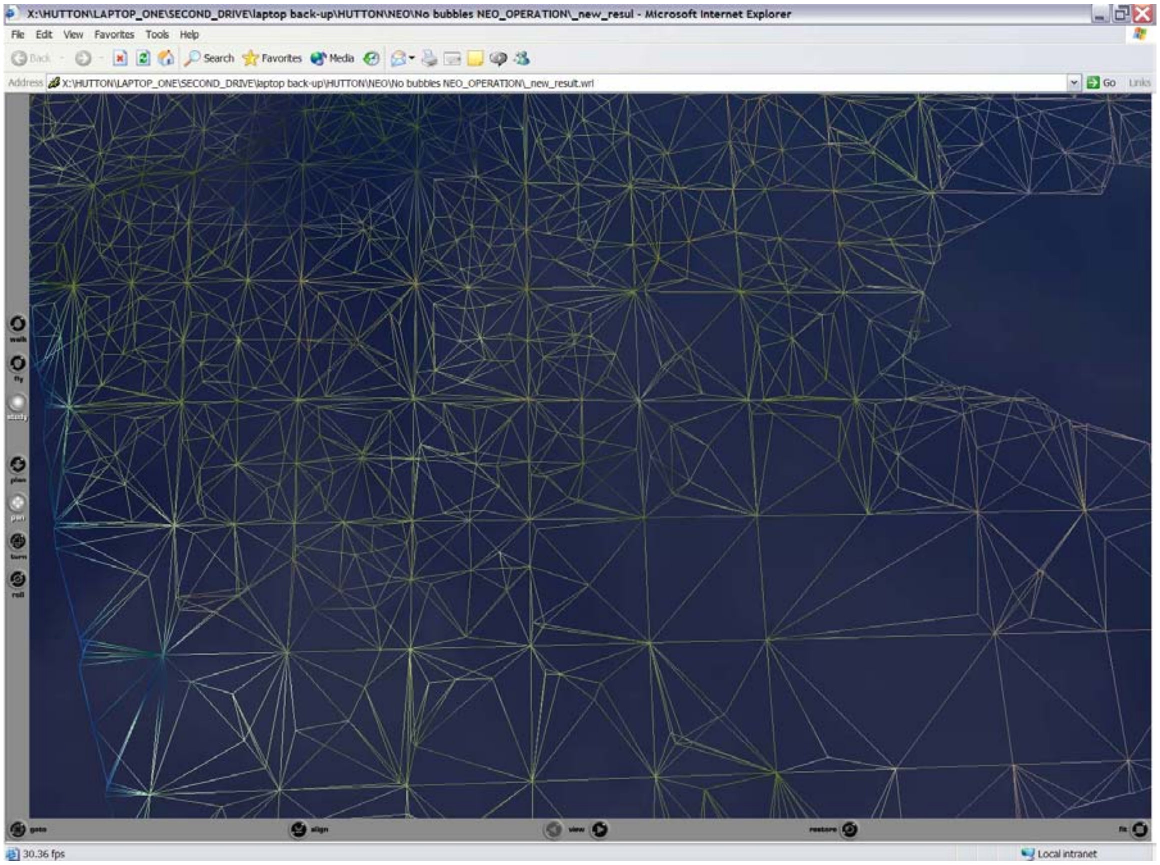

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\No bubbles NEO_OPERATION_new_resul - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\No bubbles NEO_OPERATION_new_result.wrl
Go
Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
30.36 fps
Local intranet

Figure 21. Screen shot showing polygons of the Oahu terrain using a Delaunay terrain system.

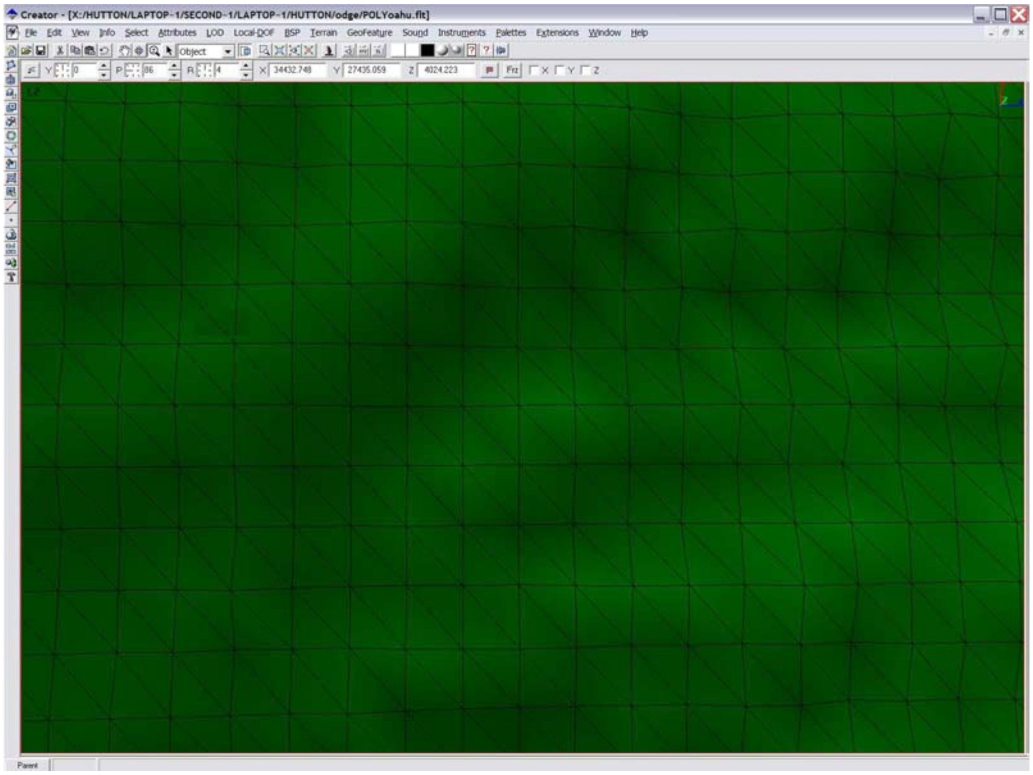

Creator - [X:/HUTTON/LAPTOP-1/SECOND-1/LAPTOP-1/HUTTON/odge/POLYoahu.flt]
File Edit View Info Select Attributes LOD Local-DOF BSP Terrain GeoFeature Sound Instruments Palettes Extensions Window Help
Object
X 34432.748 Y 27435.059 Z 4024.223
Frz □ X □ Y □ Z
Parent

Figure 22. Screen shot showing polygons of Oahu terrain using a polymesh terrain system.

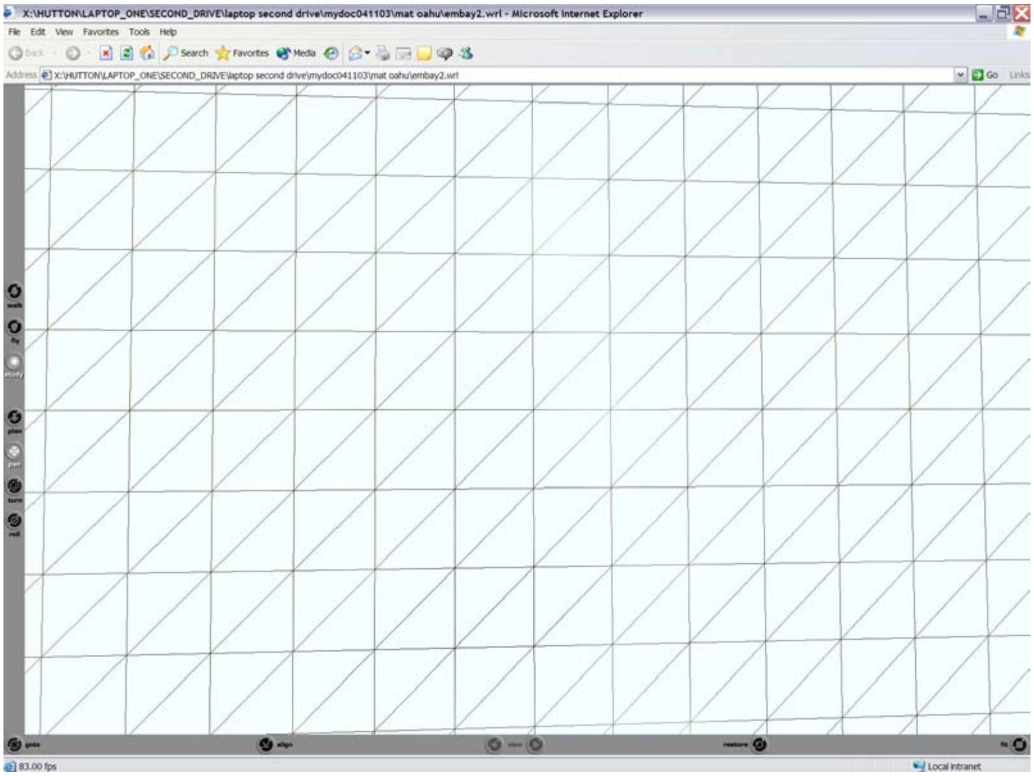
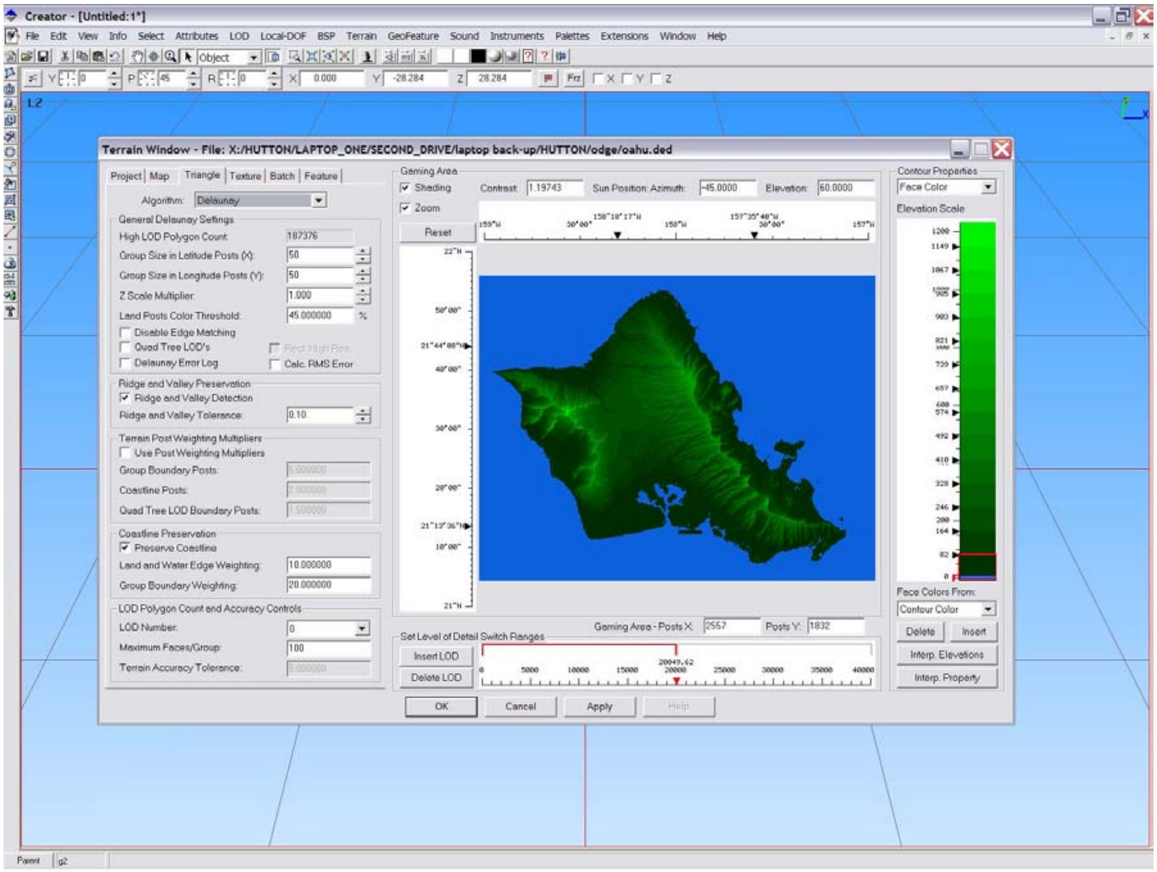

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop second drive\mydoc041103\mat oahu\embay2.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop second drive\mydoc041103\mat oahu\embay2.wrl
Go Links
walk
fly
study
plan
pan
turn
roll
goto
align
restore
fit
83.00 fps
Local intranet

Figure 23. Screen shot showing polygons of a test terrain file built from an elevation grid.

d. Building the Terrain File

After the DTED data has been read in MultiGen Creator, it builds a digital elevation data (DED) file. A DED file is a common file format used in building 3D gaming applications. Once the terrain DED file is created, the terrain file can be built. Once the DED has been read, a window appears that allows options to be chosen to determine what algorithms to apply to the DED file to generate the terrain file (Figure 24). The Oahu scene uses the Deluanay algorithm with coastline and ridge detection specified.



Y 0 P 45 R 0 X 0.000 Y -28.284 Z 28.284 □ X □ Y □ Z

Terrain Window - File: X:/HUTTON/LAPTOP_ONE/SECOND_DRIVE/laptop back-up/HUTTON/odge/oahu.ded

Algorithm: Delaunay

General Delaunay Settings
High LOD Polygon Count: 187376
Group Size in Latitude Posts (X): 50
Group Size in Longitude Posts (Y): 50
Z Scale Multiplier: 1.000
Land Posts Color Threshold: 45.000000 %
□ Disable Edge Matching
□ Quad Tree LOD's
□ Rect High Res
□ Delaunay Error Log
□ Calc. RMS Error

Ridge and Valley Preservation
☑ Ridge and Valley Detection
Ridge and Valley Tolerance: 0.10

Terrain Post Weighting Multipliers
□ Use Post Weighting Multipliers
Group Boundary Posts: 5.000000
Coastline Posts: 2.000000
Quad Tree LOD Boundary Posts: 1.500000

Coastline Preservation
☑ Preserve Coastline
Land and Water Edge Weighting: 10.000000
Group Boundary Weighting: 20.000000

LOD Polygon Count and Accuracy Controls
LOD Number: 0
Maximum Faces/Group: 100
Terrain Accuracy Tolerance: 5.000000

Gaming Area
☑ Shading Contrast: 1.19743 Sun Position: Azimuth: -45.0000 Elevation: 60.0000
☑ Zoom

Gaming Area - Posts X: 2557 Posts Y: 1832
Set Level of Detail Switch Ranges

Contour Properties: Face Color
Elevation Scale
Face Colors From: Contour Color

Figure 24. MultiGen Creator window showing the terrain building options.

e. Building the Texture for the Terrain

The next step in building the Oahu model is to build an image for the terrain data now represented as polygons. Images are called textures when used in 3D models. The texture is built for Oahu is constructed from a Compressed ARC Digitized Raster Graphics (CDRG) map from National Imagery and Mapping Agency (NIMA).

The map data is read using ArcView and exported out in a JPEG format. Then the maps are spliced together using PhotoShop. This process takes over 20 hours to complete. Once the final texture is completed, it is mapped to the terrain using MultiGen Creator (Figure 25).

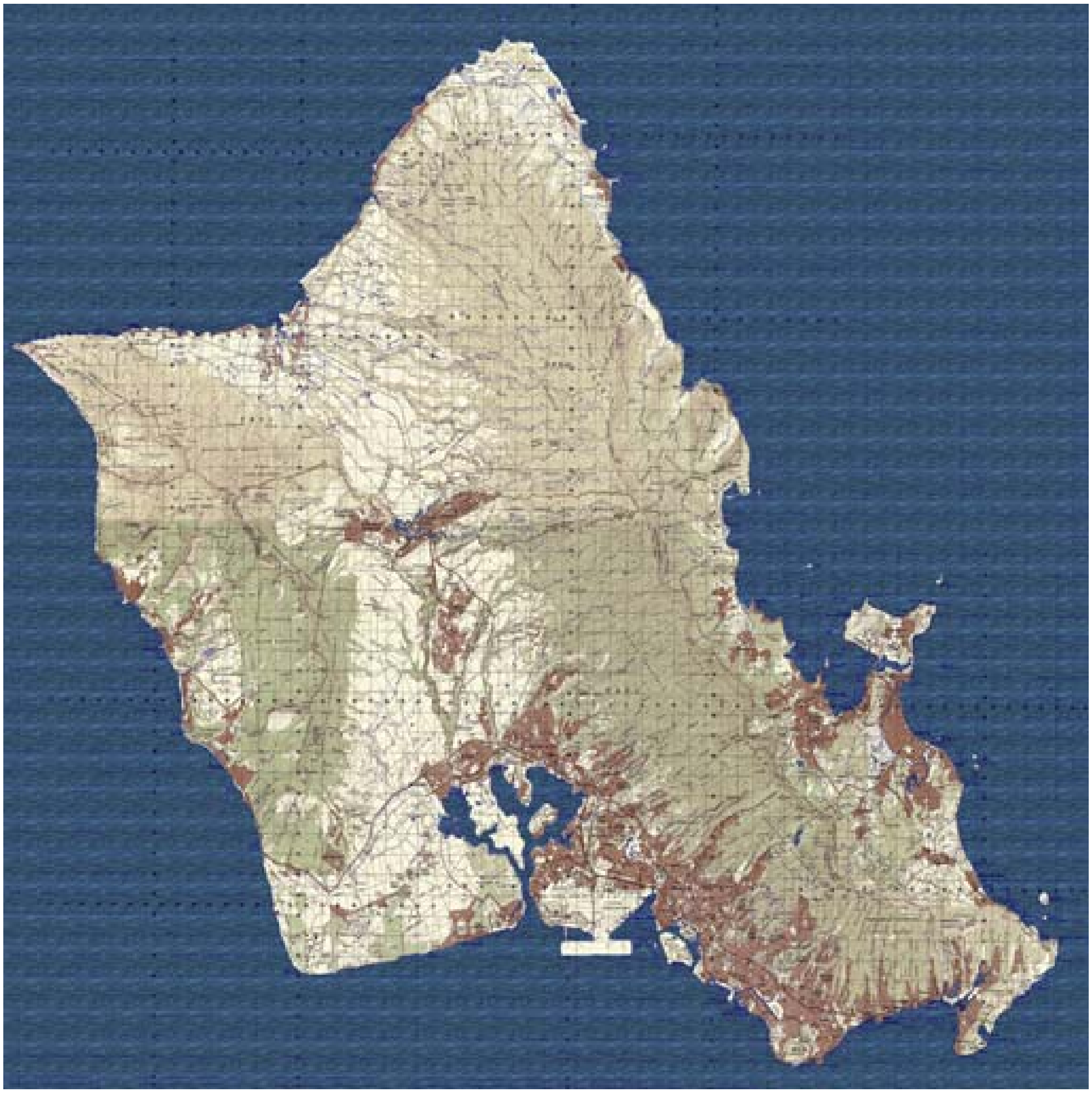
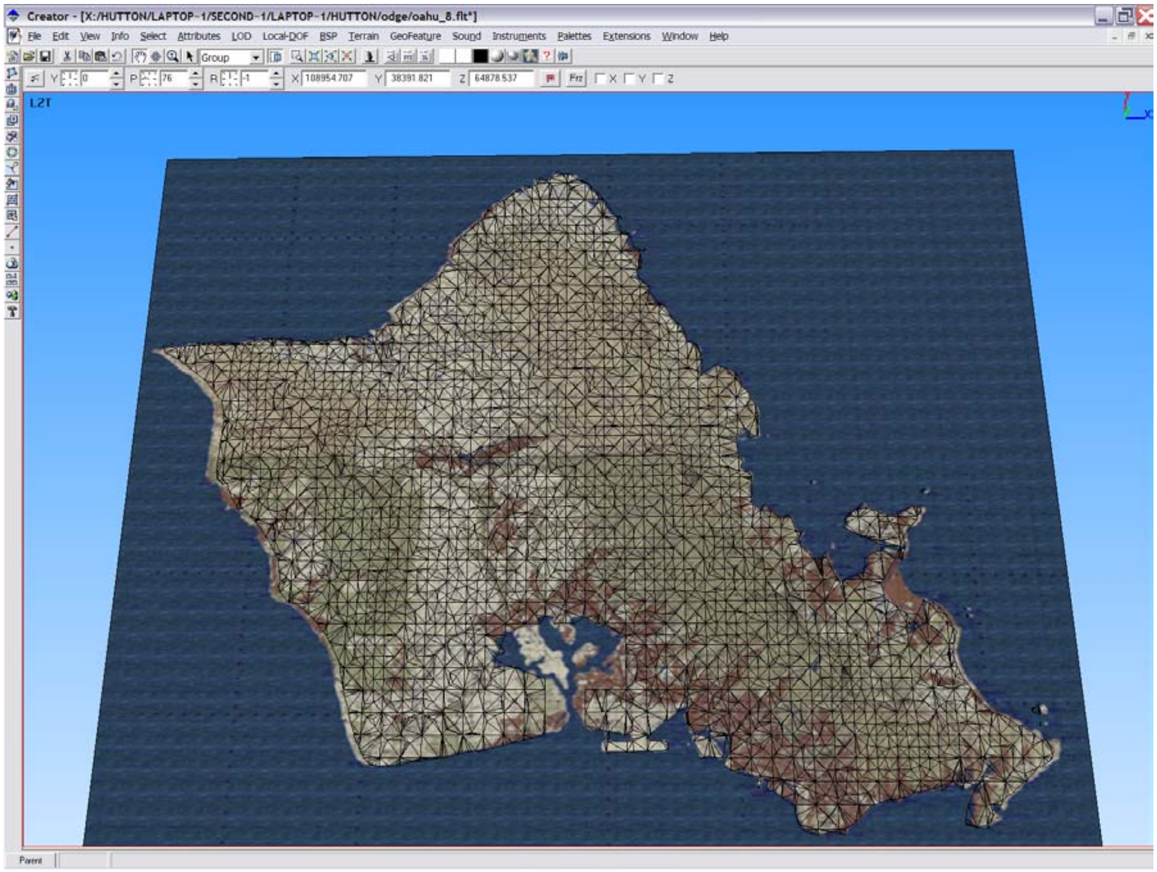


Figure 25. Completed image texture built from several 1:50,000 map sheets of Oahu.

f. Overlaying Texture Maps on the Terrain

This process is simplified by using MultiGen Creator. The user simply selects the image to be used as a texture. Then the user picks three points on the terrain.

The image gets mapped to these three points (Figure 26). The final model shows the image mapped as a texture on the terrain.


Creator - [X:/HUTTON/LAPTOP-1/SECOND-1/LAPTOP-1/HUTTON/odge/oahu_8.flt*]

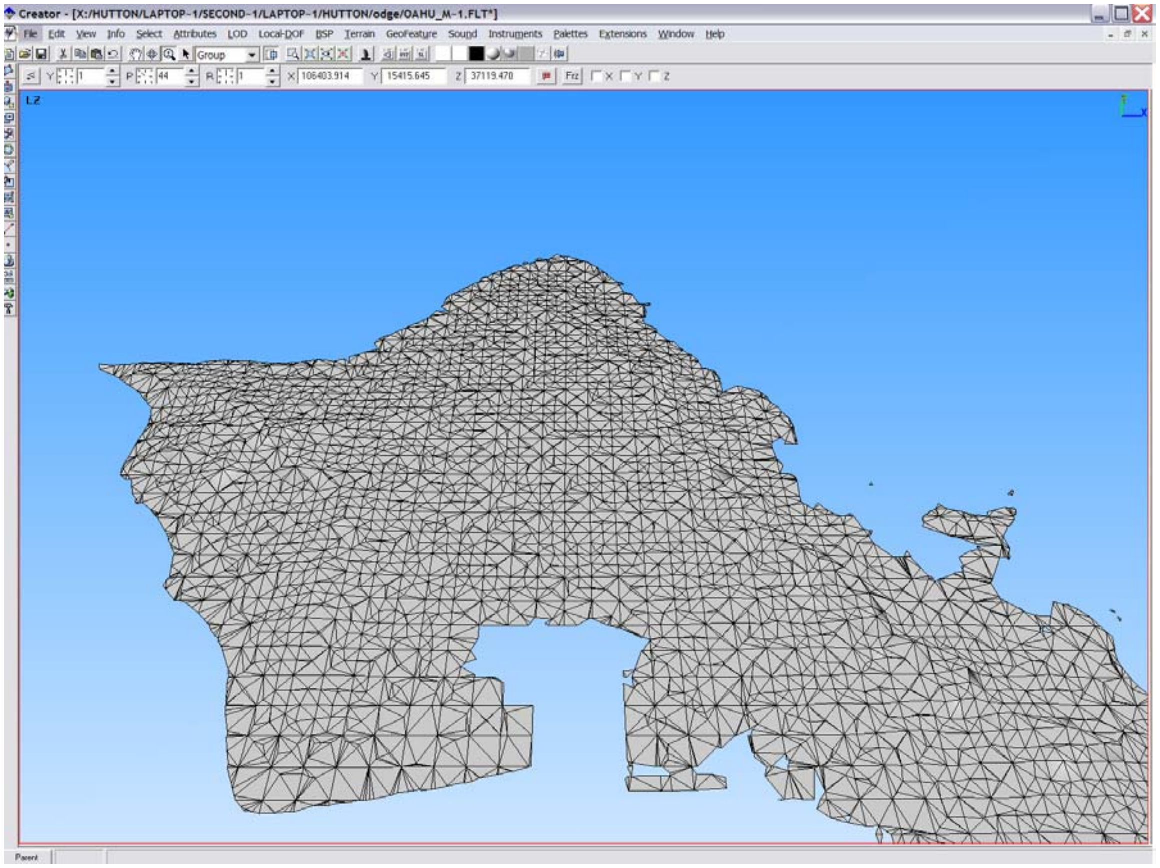
Group

Y 0 P 76 R -1 X 108954.707 Y 38391.821 Z 64878.537

Figure 26. Screen shot showing Oahu map texture being mapped onto polygons in MultiGen Creator.

g. Modifying the Terrain

Due to the requirements of other students using the terrain, modifications are made to the terrain model. The modifications involve deleting the polygons around the island representing the water. The polygons that form the mouth of the harbor and Ford Island are also deleted in the Oahu scene (Figure 27). This is done so that when ships move around on the surface they do not drive under the overlaid texture. The surface the ships moved around on is set to a single elevation value.


Creator - [X:/HUTTON/LAPTOP-1/SECOND-1/LAPTOP-1/HUTTON/odge/OAHU_M-1.FLT*]
File Edit View Info Select Attributes LOD Local-DOF BSP Terrain GeoFeature Sound Instruments Palettes Extensions Window Help
Group
Frz
□ X □ Y □ Z
LZ
Parent

Y 1 P 44 R 1 X 106403.914 Y 15415.645 Z 37119.470

Figure 27. Modified Oahu scene showing polygons removed from the harbor and Ford Island displayed in MultiGen Creator.

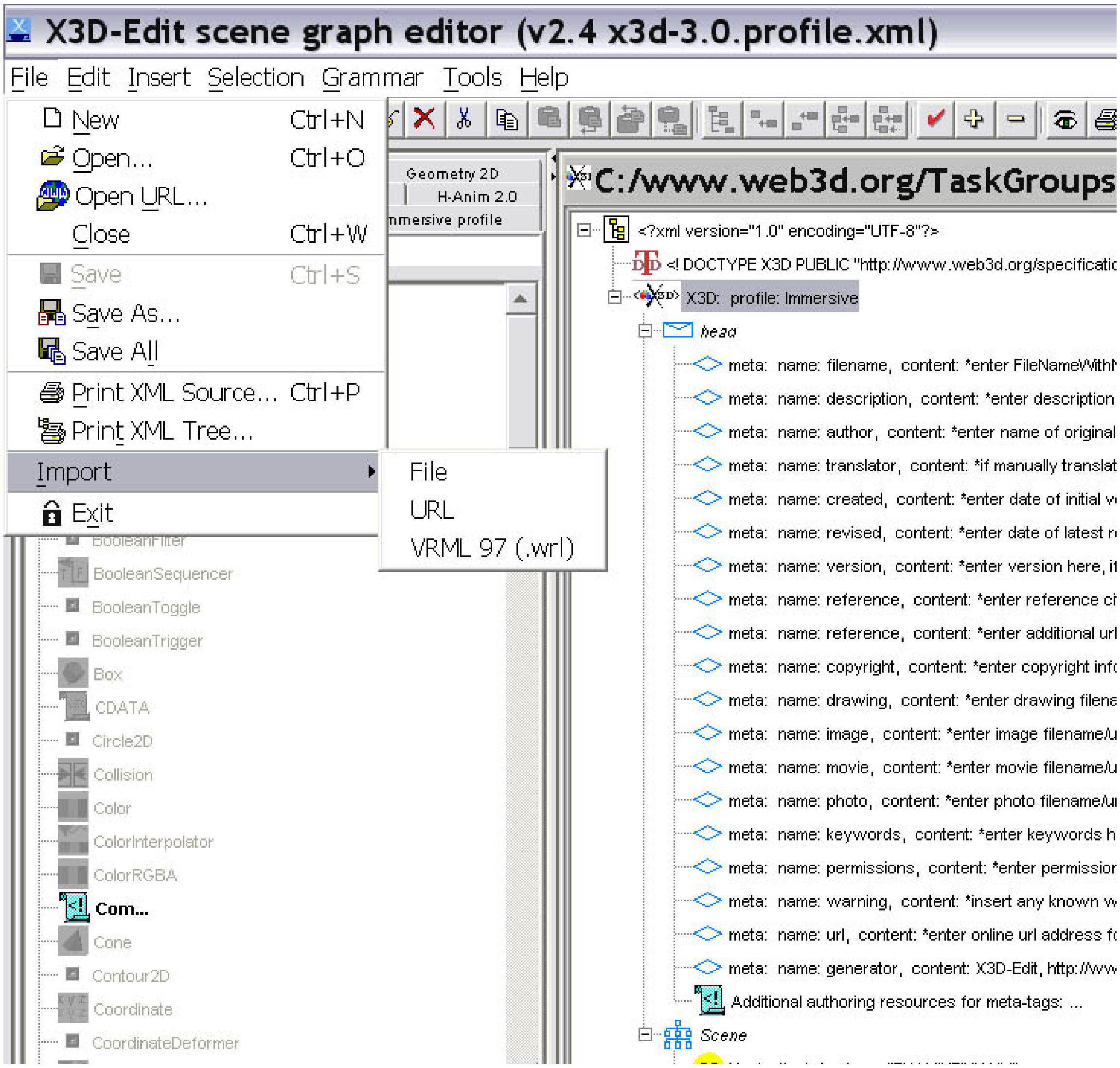
h. Conversion to VRML Format

To convert the model into a VRML format from the Open Flight format Polytrans is used. Converting the model is a very simple task with this tool and takes only a few minutes. Once the model is converted it is then pulled into Internet Model Optimizer to reduce the size of the file. Due to the size of the file this process takes about thirty minutes. The end result is a file that is just over one megabyte in size.

i. Inline of Terrain into X3D

The final step is to convert the VRML file into an X3D file. This is accomplished by using the VRML import feature found in X3D-Edit tool (Figure 28). After the file is converted into an X3D file format, it is inlined into a new X3D scene file. A 3D axis model is added to the scene to ensure correct location of the scene (Figure 29).

The scene is set up in the positive X and negative Z quadrant. The scene was later changed so that all future inlined objects in the scene would be in the positive X-Z coordinate plane.



```
C:/www.web3d.org/TaskGroups
<?xml version="1.0" encoding="UTF-8"?>
<! DOCTYPE X3D PUBLIC "http://www.web3d.org/specificatic
X3D:  profile: Immersive
  head
    meta:  name: filename,  content: *enter FileNameWithN
    meta:  name: description,  content: *enter description
    meta:  name: author,  content: *enter name of original
    meta:  name: translator,  content: *if manually translat
    meta:  name: created,  content: *enter date of initial v
    meta:  name: revised,  content: *enter date of latest r
    meta:  name: version,  content: *enter version here, if
    meta:  name: reference,  content: *enter reference ci
    meta:  name: reference,  content: *enter additional url
    meta:  name: copyright,  content: *enter copyright infc
    meta:  name: drawing,  content: *enter drawing filena
    meta:  name: image,  content: *enter image filename/u
    meta:  name: movie,  content: *enter movie filename/u
    meta:  name: photo,  content: *enter photo filename/ur
    meta:  name: keywords,  content: *enter keywords h
    meta:  name: permissions,  content: *enter permissior
    meta:  name: warning,  content: *insert any known w
    meta:  name: url,  content: *enter online url address fc
    meta:  name: generator,  content: X3D-Edit, http://ww
    Additional authoring resources for meta-tags: ...
  Scene
```

Figure 28. Screen shot of X3D–Edit tool showing import feature.

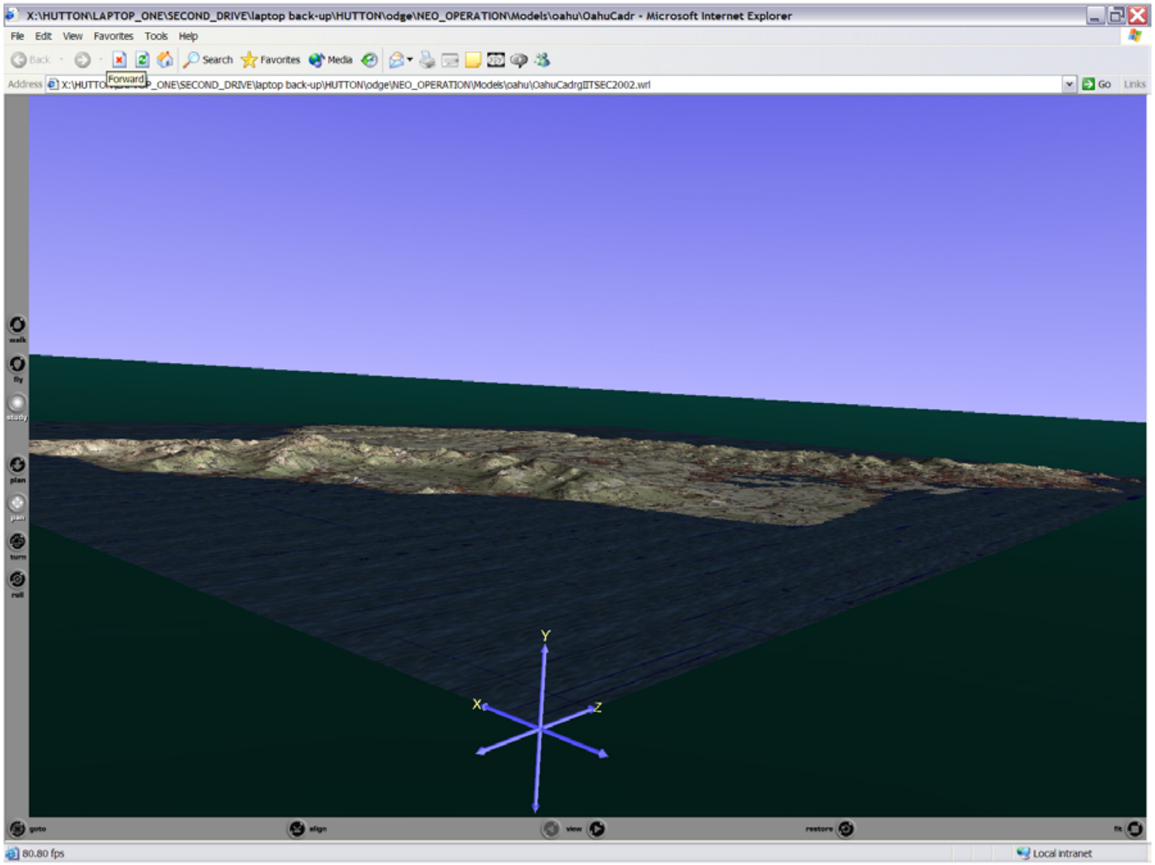

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\odge\NEO_OPERATION\Models\oahu\OahuCadr - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Forward
Search
Favorites
Media
Address
X:\HUTTO_P_ONE\SECOND_DRIVE\laptop back-up\HUTTON\odge\NEO_OPERATION\Models\oahu\OahuCadrgIITSEC2002.wrl
Go
Links
walk
fly
study
plan
pan
turn
roll
Y
X
Z
goto
align
view
restore
fit
80.80 fps
Local intranet

Figure 29. Final Oahu scene with axis.

j. Placing Objects on the Terrain

With the terrain now place in the correct location new objects can be added to the scene. The approach to adding objects is similar to the process used in the NPS model scene with two exceptions. The first is that since a map texture is used the scene map grid system is used to get the X and Z coordinates. The second change is in the Y coordinate system. Because the terrain is not flat adding objects takes more time. A bracketing system is used to find the correct Y-coordinate location. This usually takes about four to five tries to find a good location for an object.

k. Using Oahu Model for a Non-Combatant Evacuation Operation (NEO)

Now that an area of operation is developed a military operation is modeled. A NEO scenario is used for the model. Developing the NEO scenario involves picking locations like coastal landing beaches (CLB), helicopter landing zones (HLZ), airfields, ports, evacuation sites, and the location of a United States Embassy and other

U.S. facilities. Once these locations are chosen, they are represented on the map using different colored dome objects (Figure 30). The U.S. Embassy is represented using the NPS campus model (Figure 31).

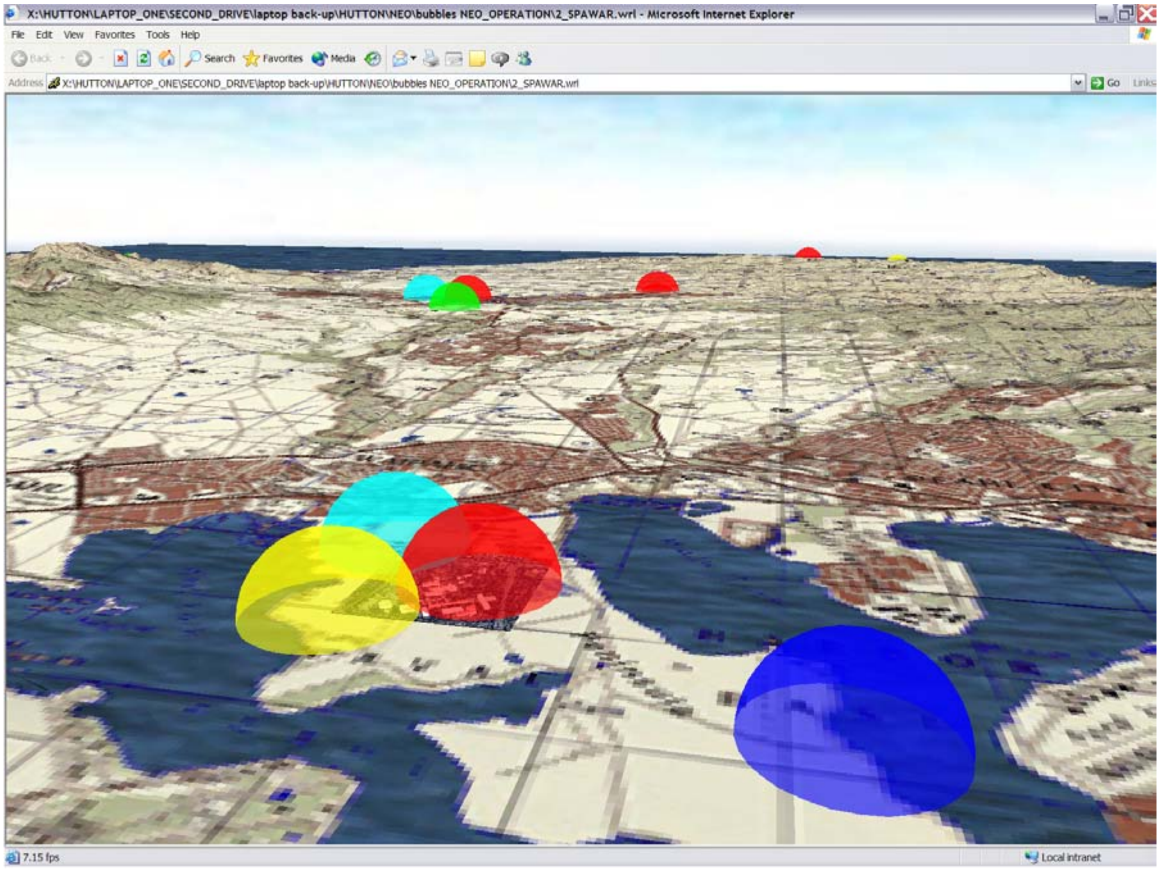

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl
Go Links
7.15 fps
Local intranet

Figure 30. Screen shot from Oahu scene showing use of domes to represent area of interest.

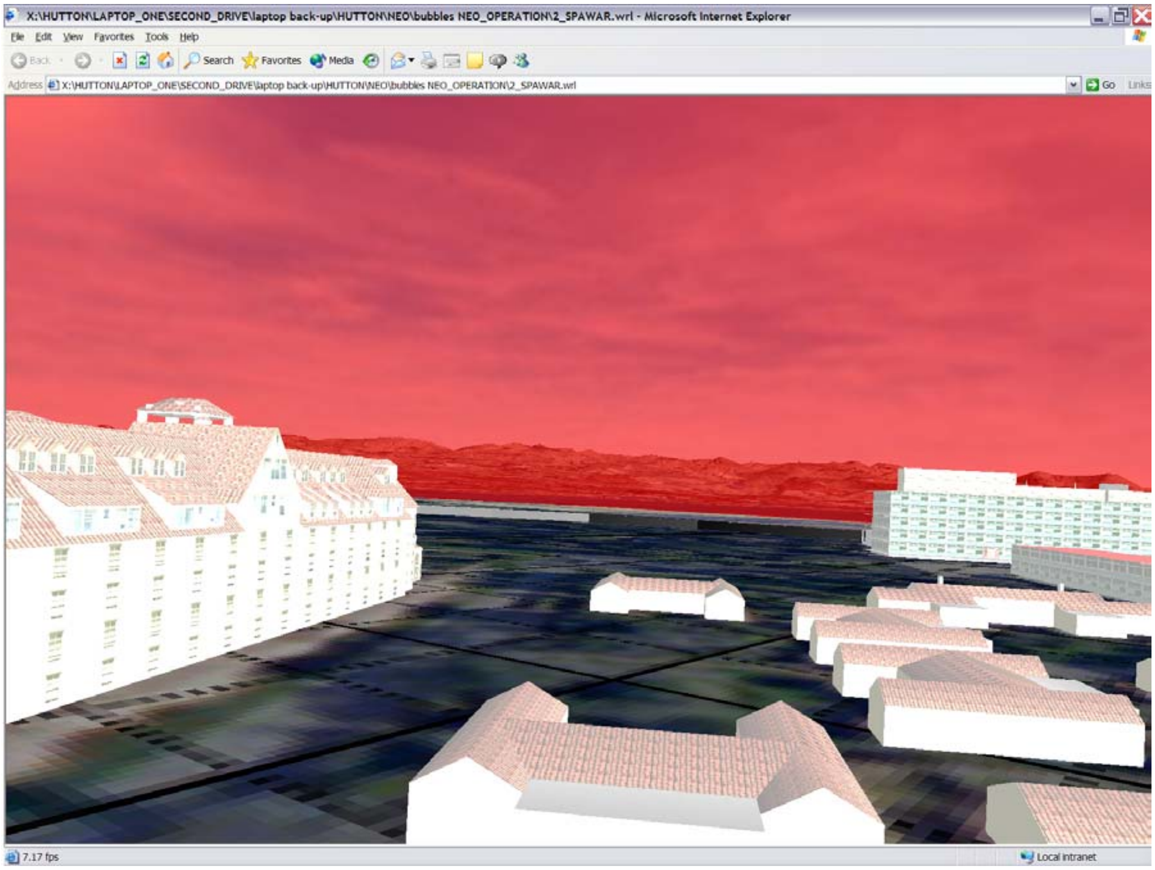

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl
Go
Links
7.17 fps
Local intranet

Figure 31. Screen shot from NPS campus in Oahu scene.

l. Use of Ships, Aircraft, and Other Military Models

To complete the model of the NEO an LHD Wasp class ship and military aircraft are used (Figure 32). The LHD Wasp class ship is obtained from the 3D model library at https://models.tec.army.mil/models/home.html (accessed August 2003). The model required modification since it was in an Open Flight format and represented an Iwo Jima class ship. A Cobra attack helicopter is obtained from the Army Model Exchange at https://modelexchange.army.mil/amx_public/amxPub_main.asp (accessed August 2003). The model also had to be converted into a VRML format and required minor modifications. Lastly a V22 model is obtained at NPS along with amphibious assault vehicles (AAV) and a Harrier jet. [Savage 2003] [Nicklaus 2001]

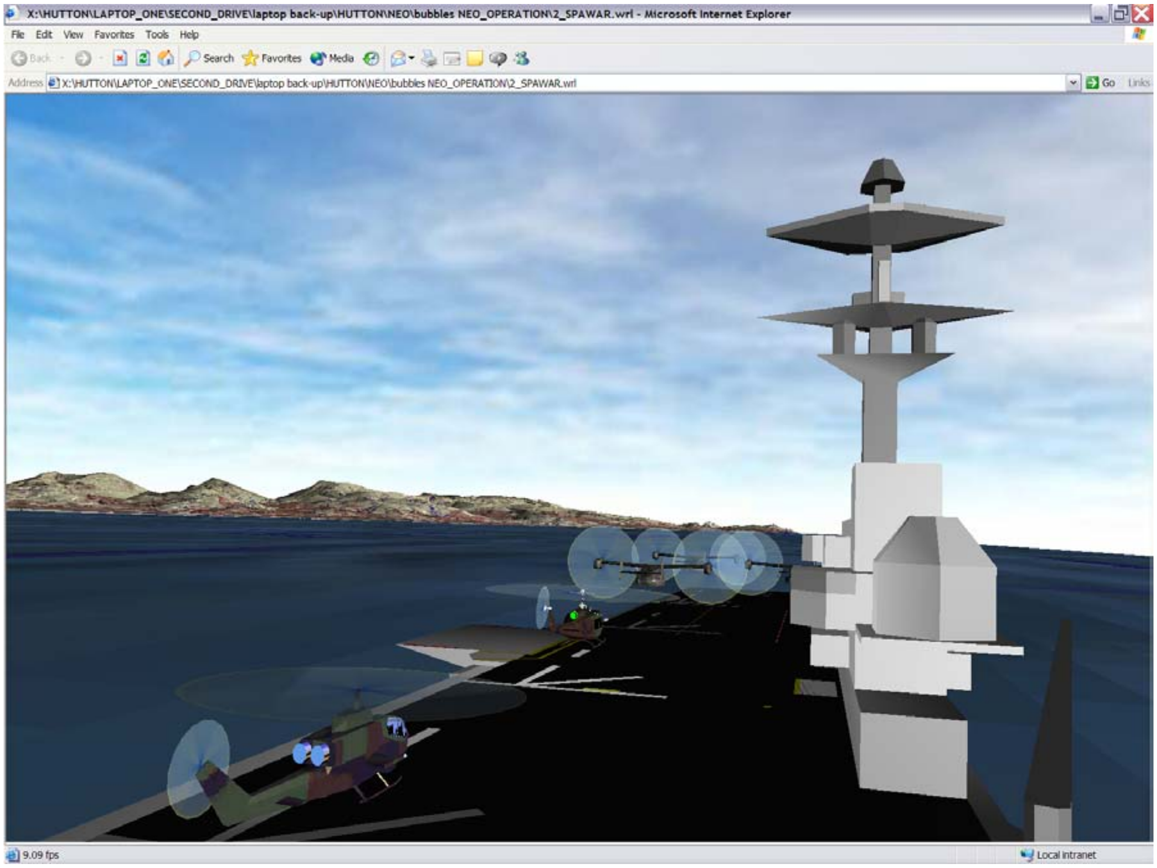

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\bubbles NEO_OPERATION\2_SPAWAR.wrl
Go
Links
9.09 fps
Local intranet

Figure 32. Screen shot of military models used in Oahu scene.

m. Use of Waypoint Interpolator Prototype

In order to simulate the NEO in action, a X3D prototype is used to add realistic movement to the military vehicles conducting the NEO. The Waypoint Interpolator prototype allows the user to pass the X Y Z locations and the speed that an object will move. Then the code generates the movement that is desired. The final result is a representation of a NEO operation occurring on the island of Oahu (Figure 33). The Waypoint Interpolator prototype can be found at the following URL: http://web.nps.navy.mil/~brutzman/Savage/Tools/Animation/chapter.html (accessed August 2003).

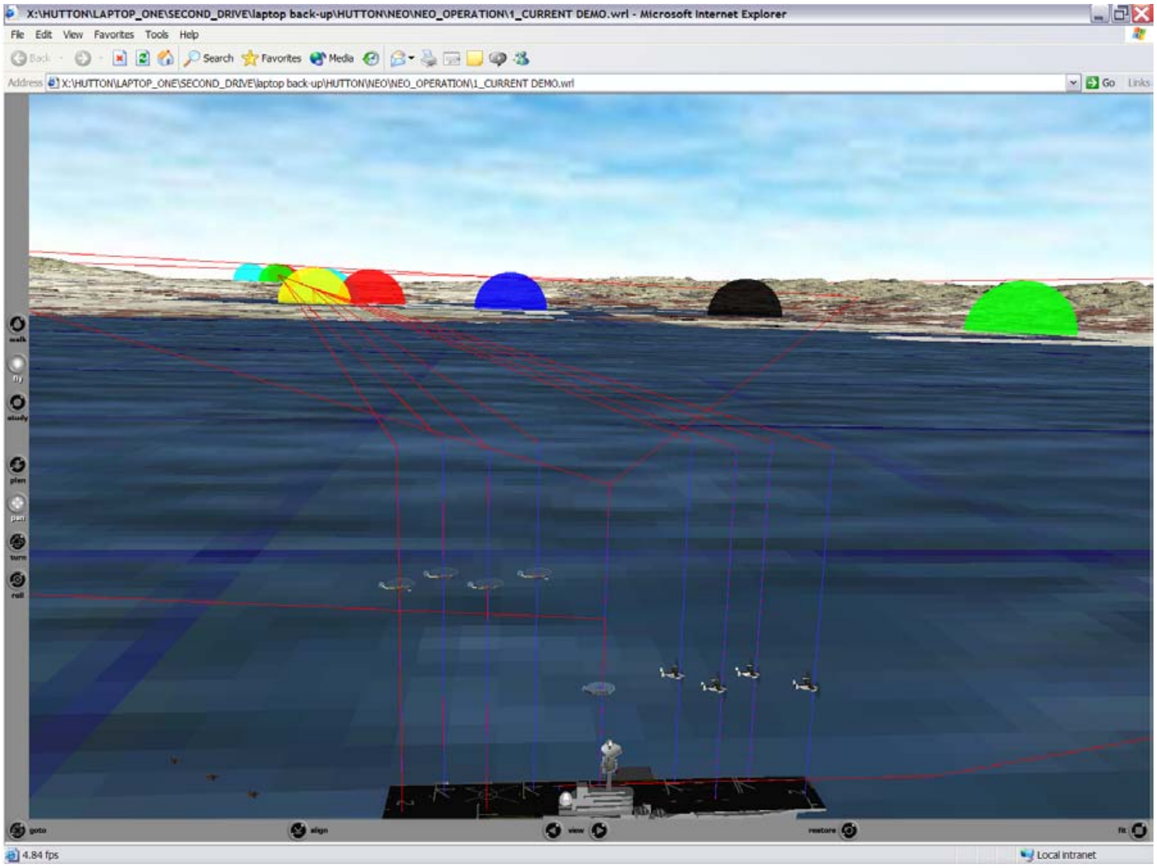
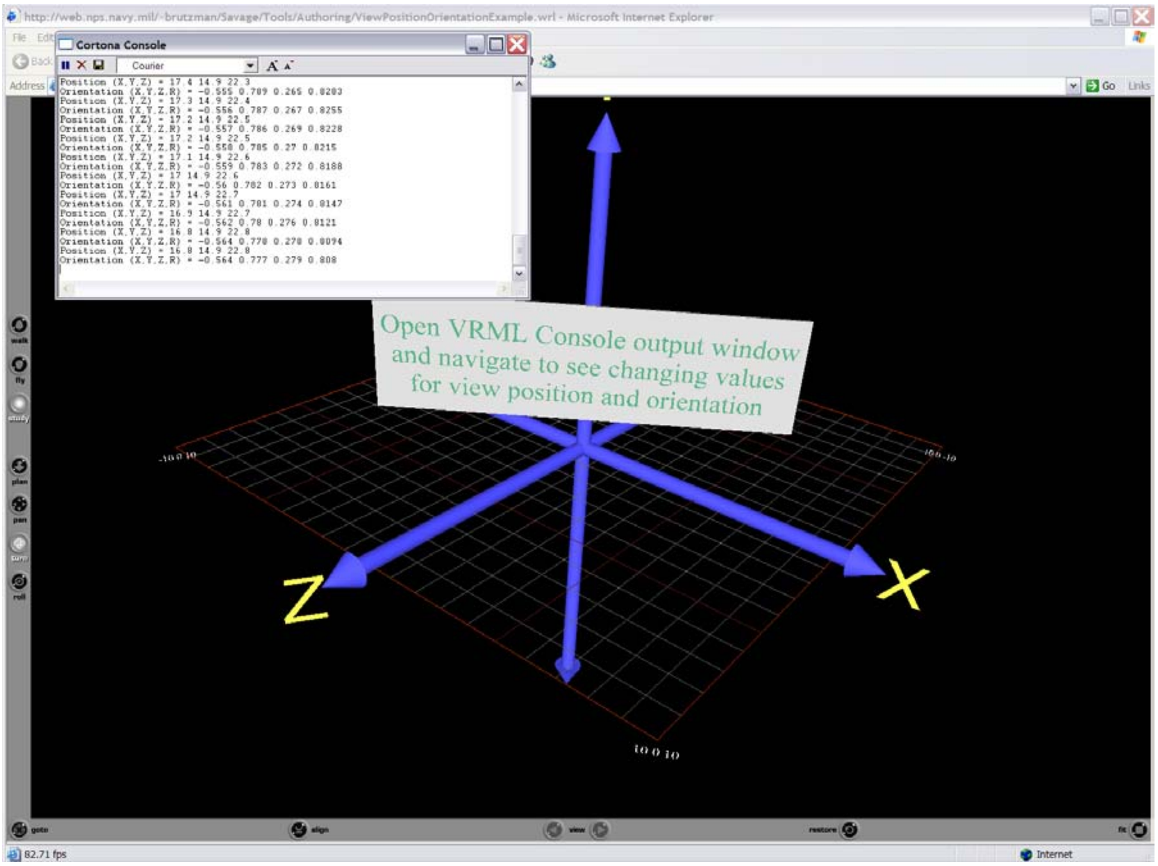

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\NEO_OPERATION\1_CURRENT DEMO.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\NEO\NEO_OPERATION\1_CURRENT DEMO.wrl
Go Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
4.84 fps
Local intranet

Figure 33. Simulated Non-Combatant Evacuation Operation in Oahu scene demonstrating use of waypoint interpolator to move aircraft.

n. Use of View Position Orientation Prototype

The View Position Orientation prototype is a X3D prototype used to help develop the NEO model. This VRML prototype outputs the location and orientation that the user is at while in a 3D VRML scene (Figure 34). This is helpful in two ways during development. First it allows the user to develop view points for the scene. Another common use is to help determine the X Y Z locations needed for the waypoint interpolator. The view position orientation can be found at the following URL: http://web.nps.navy.mil/~brutzman/Savage/Tools/Authoring/chapter.html (accessed August 2003).



http://web.nps.navy.mil/~brutzman/Savage/Tools/Authoring/ViewPositionOrientationExample.wrl - Microsoft Internet Explorer
Cortona Console
Courier
Open VRML Console output window
and navigate to see changing values
for view position and orientation
Z
X
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
82.71 fps
Internet

```
Position (X,Y,Z) = 17.4 14.9 22.3
Orientation (X,Y,Z,R) = -0.555 0.789 0.265 0.8283
Position (X,Y,Z) = 17.3 14.9 22.4
Orientation (X,Y,Z,R) = -0.556 0.787 0.267 0.8255
Position (X,Y,Z) = 17.2 14.9 22.5
Orientation (X,Y,Z,R) = -0.557 0.786 0.269 0.8228
Position (X,Y,Z) = 17.2 14.9 22.5
Orientation (X,Y,Z,R) = -0.558 0.785 0.27 0.8215
Position (X,Y,Z) = 17.1 14.9 22.6
Orientation (X,Y,Z,R) = -0.559 0.783 0.272 0.8188
Position (X,Y,Z) = 17 14.9 22.6
Orientation (X,Y,Z,R) = -0.56 0.782 0.273 0.8161
Position (X,Y,Z) = 17 14.9 22.7
Orientation (X,Y,Z,R) = -0.561 0.781 0.274 0.8147
Position (X,Y,Z) = 16.9 14.9 22.7
Orientation (X,Y,Z,R) = -0.562 0.78 0.276 0.8121
Position (X,Y,Z) = 16.8 14.9 22.8
Orientation (X,Y,Z,R) = -0.564 0.778 0.278 0.8094
Position (X,Y,Z) = 16.8 14.9 22.8
Orientation (X,Y,Z,R) = -0.564 0.777 0.279 0.808
```

Figure 34. Screen shot showing an example of the View Position Orientation prototype.

o. Conversion of Oahu Model to X3D Geospatial Profile

A few months after the scene was developed it was converted into an X3D Geospatial Profile scene. This process involves adding a geo-origin node and the required X3D Geospatial Profile prototypes into the scene. This now allows objects to be added into the scene based upon a geo-location using latitude, longitude and altitude.

p. Current Status of Model

The model is now in X3D and VRML formats and is usable for future studies. Several versions of the model currently exist. Future development can continue with this model.

3. Country of Iraq

a. Obtaining DTED Data

Level 1 DTED data is used for the country of Iraq model. Obtaining the DTED data for Iraq was more involved than getting data for Oahu. The reason was the data is located on four different DTED CDs. The individual data files that covered Iraq on each CD must be found and placed into one directory on the computer. The process takes about two hours. After the data is in one location, then the same procedures used to read the data for Oahu are used.

b. Building the Terrain File

The terrain file for Iraq is built using MultiGen Creator. The settings used Deluanay and ridge and valley detection set. The resulting terrain file appears similar to a polymesh terrain file.

c. Using a Different Map Projection

One modification in building the Iraq terrain is the use of a different map projection option in the Multigen Creator tool. The type of map projection is geocentric. The earth ellipsoid model uses WGS84. This gives the appearance of the curved surface of the earth (Figure 35).

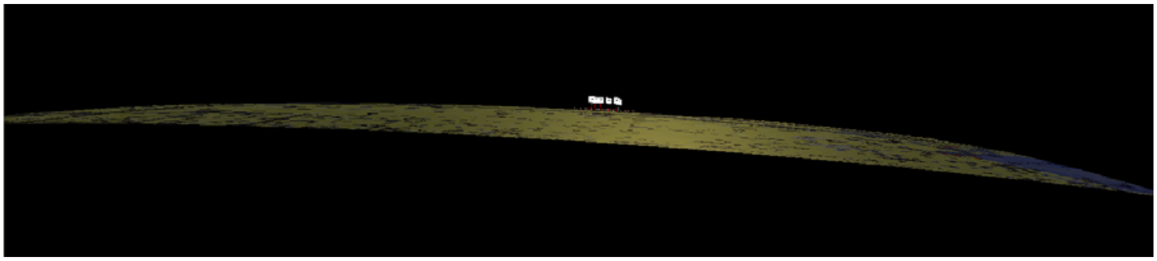


Figure 35. Iraq Scene showing curved terrain.

d. Tiling Terrain

One common method used to build large areas of terrain is to create a tile system. In order to allow for future development, a tile system is also used for the Iraq terrain. In the MultiGen Creator tool, the user can define the size and number of tiles desired for a given area. For the Iraq terrain twenty five tiles, each two degrees by two degrees, are built (Figure 36).

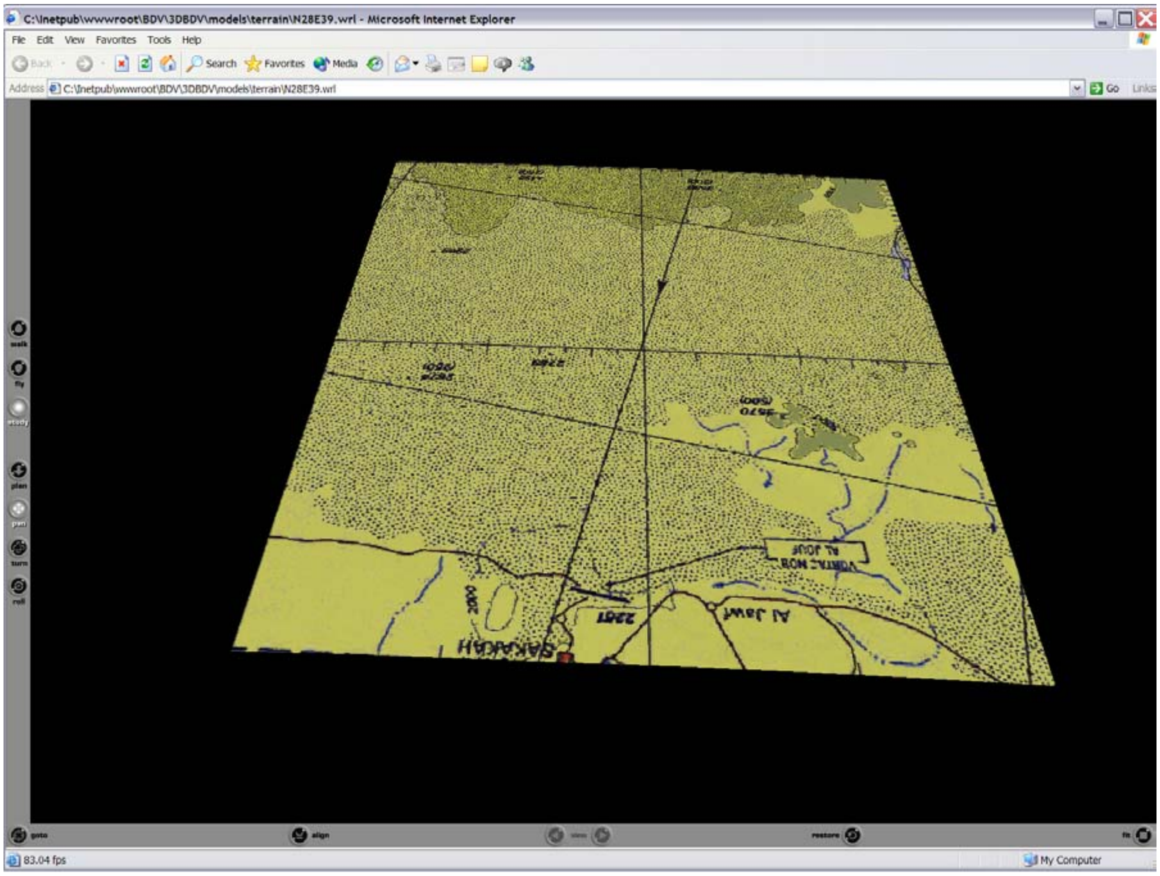

C:\Inetpub\wwwroot\BDV\3DBDV\models\terrain\N28E39.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Inetpub\wwwroot\BDV\3DBDV\models\terrain\N28E39.wrl
Go Links
walk
fly
study
plan
pan
turn
roll
Al Jawf
2351
goto
align
view
restore
fit
83.04 fps
My Computer

Figure 36. Screen shot of one of the 25 tiles used in the Iraq scene. Each tile represented an area 2 degree in latitude by 2 degree in longitude.

e. Building the Textures for the Terrain

CADRG maps from NIMA are used as the textures for the Iraq terrain. Twenty-five map textures are built in order to cover the 25 terrain files that cover Iraq. The textures are built in the same manner used to build the texture for Oahu. However, the textures are not combined into one single texture.

f. Overlaying Texture Maps on the Terrain Tiles

This is accomplished by opening each terrain file and mapping the texture in MultiGen Creator. Unlike the Oahu terrain this process is performed 25 times. When looking at the final VRML model, it gives the appearance of just one single texture (Figure 37).

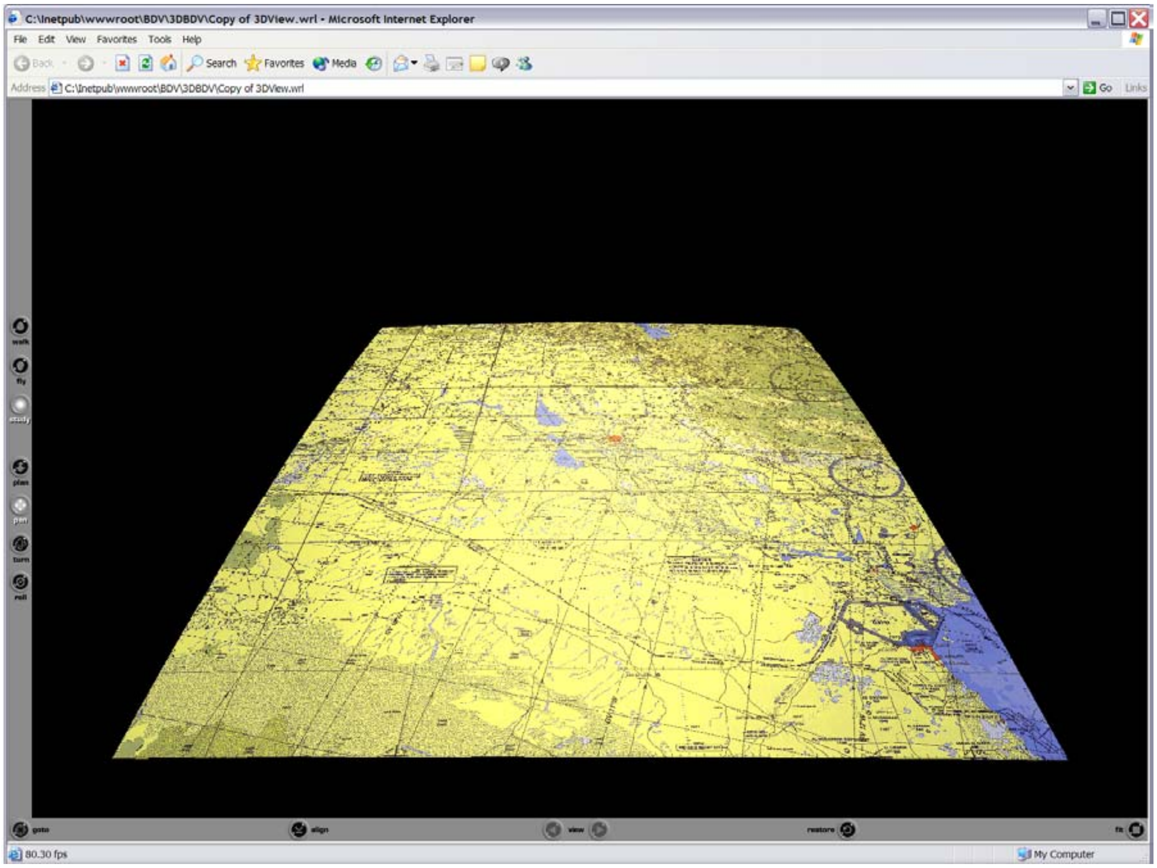

C:\Inetpub\wwwroot\BDV\3DBDV\Copy of 3DView.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address C:\Inetpub\wwwroot\BDV\3DBDV\Copy of 3DView.wrl
Go Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
80.30 fps
My Computer

Figure 37. Iraq scene showing the 25 individual tiles together.

g. Conversion to VRML Format

Before the model is converted into a VRML format, it needs to be repositioned in MultiGen Creator. Because the Geocentric system is used the terrain is in a position that reflects its location on an actual earth sphere. In order to work with terrain, it is moved so that the lower left corner of the terrain is at the (0 0 0) location in the XYZ coordinate system. After this is accomplished, each individual tile is then converted to VRML.

h. Conversion of Model to X3D Geospatial Profile

Once the tiles are converted to VRML, the X3D Geospatial Profile code is added to each tile's file. Since the lower left tile's corner is set at (0 0 0), that location is set as the geo-origin for each tile. The same X3D Geospatial Profile code appears at the top of each file. This code is provided in Appendix A.

i. Inline of Terrain Tiles in X3D Geospatial Profile

A main VRML file is created that defines the geo-origin and each tile is then inlined into the X3D Geospatial Profile main file. Because each file now has X3D Geospatial Profile code, the tiles appear in the correct location. The final result gives the appearance of a single terrain file. By inlining each tile, future modification and research can occur.

j. Placing Objects on the Terrain

Since an X3D Geospatial Profile system is used, a normal inline to place objects can not be used. The X3D Geospatial Profile code allows objects to be placed based upon a latitude, longitude and elevation coordinate system (Figure 38). The X3D Geospatial Profile node used to accomplish this is the geo-Location prototype. This node converts the latitude, longitude and elevation coordinates into the X Y Z coordinate system used in X3d.

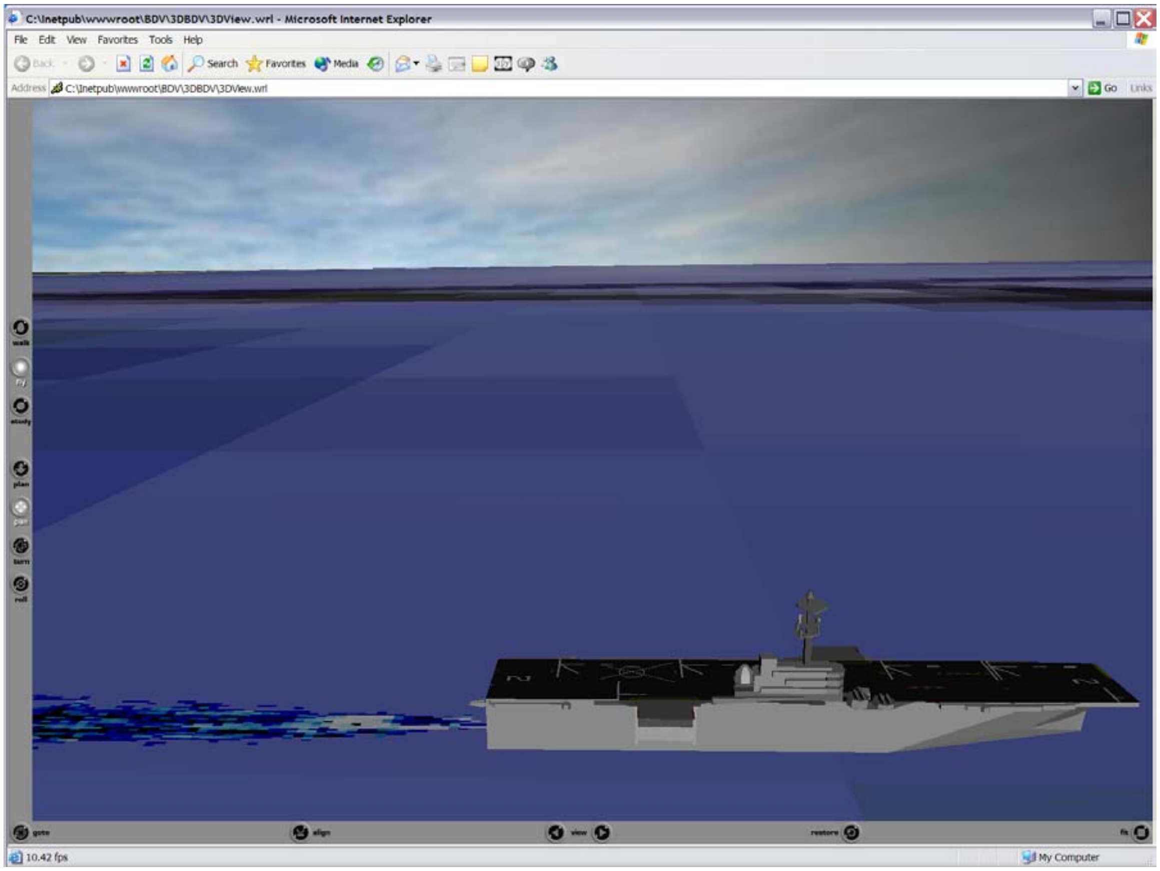

C:\Inetpub\wwwroot\BDV\3DBDV\3DView.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address C:\Inetpub\wwwroot\BDV\3DBDV\3DView.wrl
Go
Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
10.42 fps
My Computer

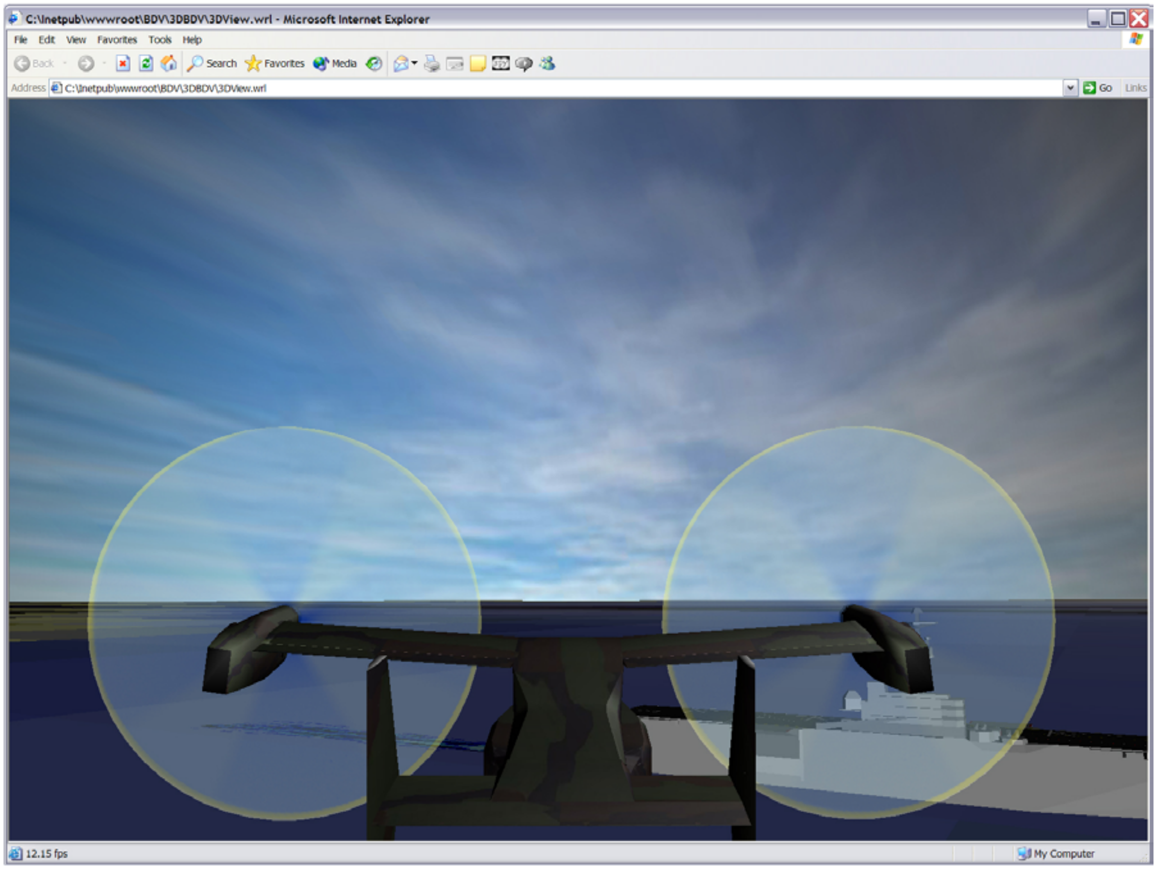
Figure 38. Iraq scene showing placement of WASP class ship using geo-location prototype.

k. Using Joystick for Navigation

Navigation in a VRML scene can be difficult if a large model is used. The Cortona VRML plug-in requires the user to select between different options for movement. The first option is the walk mode. The walk mode allows the user to walk along the surface of the scene at about eye level. The next option is the study mode. The study mode allows the user to move the scene around while keeping the user in a central location. The last mode is the fly mode. This fly mode allows the user to fly throughout the scene while the scene stays in one location. The user selects the speed that movement can occur to help control movement. Even with these tools navigation is a challenge. While conducting research a VRML prototype was found that allows a user to use an external device in order to navigate in a 3D VRML scene. The VRML device handler is found on the geovrml.com website and was developed by Eric Maranne. [Maranne 2003] For the Iraq model a joystick is used to move around within the scene. This makes navigation very easy to accomplish. An aircraft is used and the navigation prototype code now allows the user to fly around in the scene as if he is flying an aircraft.

l. Development of Aircraft for Realistic Flight

The VRML device handler code was modified to reflect similar controls that would be used to fly an aircraft. See Appendix B for the flight control code. The device handler allows the programmer to develop the code for certain actions on a joystick. For the V22 used in the Iraq model, kinematic flight controls are mapped to the joystick controls (Figure 39).



Address C:\Inetpub\wwwroot\BDV\3DBDV\3DView.wrl

12.15 fps

Figure 39. V22 aircraft model in flight using joystick to control in the Iraq scene.

Further coding was performed by Major David “Fuzzy” Wells, USAF, on the device handler code. He added the capability for different flight control surfaces of the aircraft to move based upon joystick actions. He used a Navy Catalina aircraft to demonstrate the capability of the code (Figure 40). He also mapped buttons on the joystick to raise and lower landing gear and control view points on the aircraft.

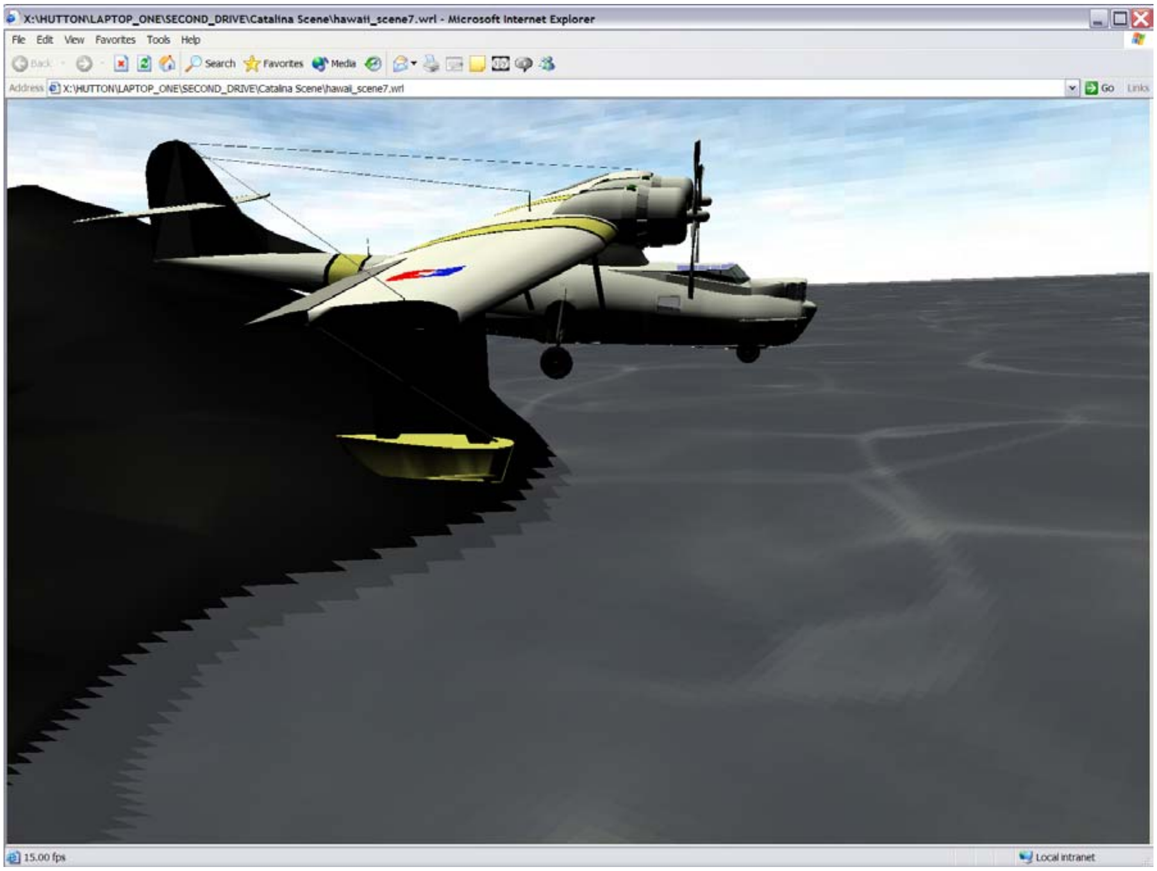

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\Catalina Scene\hawaii_scene7.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\Catalina Scene\hawaii_scene7.wrl
Go Links
15.00 fps
Local intranet

Figure 40. Catalina model developed by David "Fuzzy" Wells, USAF.

m. Use of X3D Geospatial Profile Node Inline

One prototype in the X3D Geospatial Profile is the Inline node. This prototype dynamically loads and unloads an object into a X3D scene and the memory of the user's system. When converting a model back to the VRML 97 specification, the GeoVRML 1.0 node GeoInline is used. This prototype is used with a proximity sensor to load and unload a tile in the Iraq scene. Since the Iraq scene is able to load with all tiles and still have good performance, this node was only tested and not retained in the final version of the scene.

n. Current Status of Model

Currently the Iraq model is being used for continued research with modeling of relational data in a 3D scene. The model can be used for other research needing to cover a large geographic area. The model covers 10 degrees latitude by 10 degrees longitude. Future development can continue with this model (Figure 41).

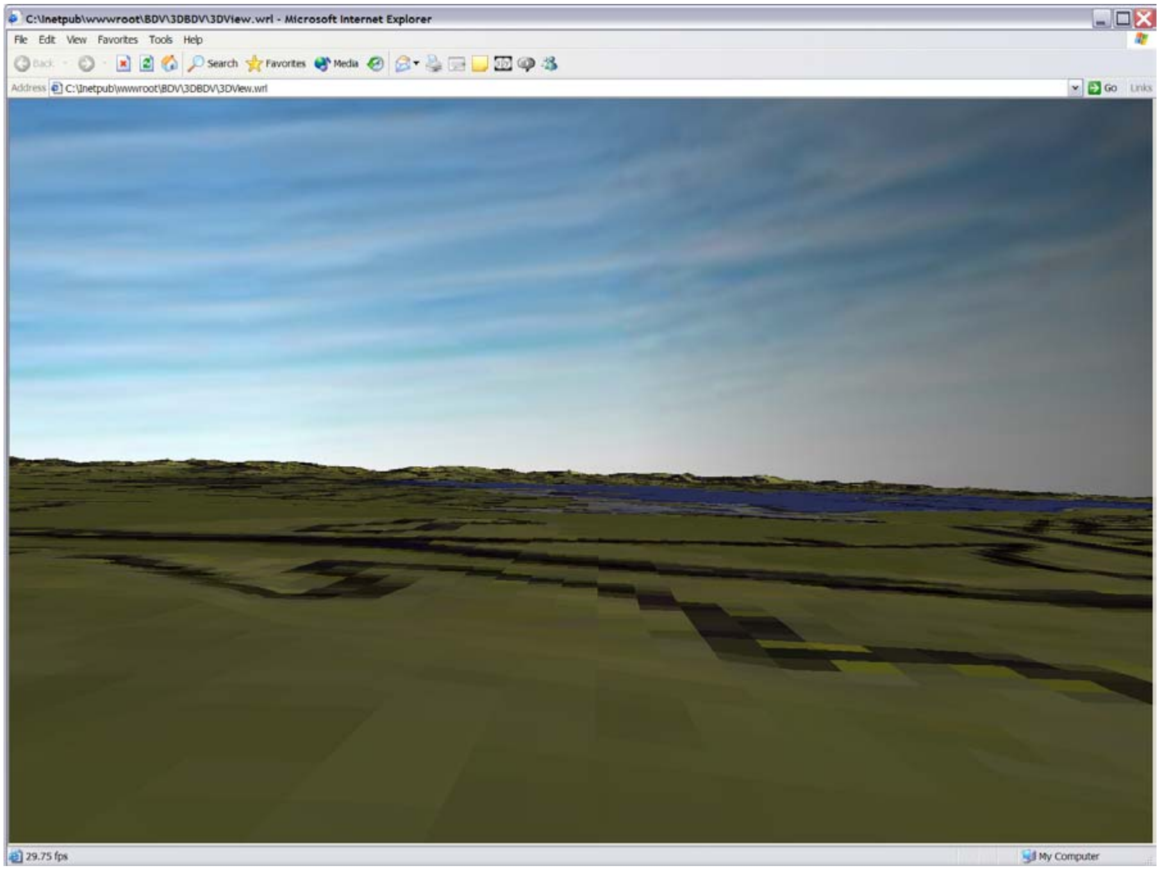

C:\Inetpub\wwwroot\BDV\3DBDV\3DView.wrl - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
C:\Inetpub\wwwroot\BDV\3DBDV\3DView.wrl
Go
Links
29.75 fps
My Computer

Figure 41. Screen shot taken from the northern area of Iraq looking southwest. This screen shot shows the mountain ranges in the northeast of Iraq.

4. Recommendations and Conclusions

a. Overview

The development of these two models took several months. They serve as the basis for further research in the thesis. With the successful development of these models, work is then able to be conducted in bringing in data from a relational database as 3D models incorporated into a large-scale 3D scene.

b. Recommendations for Further Development of the Oahu Model

The Oahu model can serve as a good foundation for continued research with modeling of relational data and 3D visualization. The drastic elevation changes and ocean surfaces allow the model to be used in a wide variety of research. Adding bathymetry data to this model is recommended. Additional work with the NEO scenario can also be continued. Having operational data added to the scene after a user selects options from a web page can be integrated into the model.

c. Recommendations for Further Development of the Iraq Model

Work with this model should continue. If needed, the model can be redone in a flat earth configuration. Additional work with the X3D Geospatial Profile prototype can be continued since the overall model is in a tile configuration. Research on terrain-following prototypes might also work well with this terrain. Combining the Baghdad Model with this model can be considered in the near future.

d. Conclusions

The work on these two models creates a great deal of learning experience and basis to facilitate further research. The development of these models is the first step in modeling operations data. Both models can be used for future research.

E. 3D MODELING OF RELATIONAL DATA RETRIEVED FROM DATABASES

1. Choosing a Database

The goal of using a database in this project is to demonstrate how operational data can be modeled in a 3D scene. For this project two types of databases are considered. The first type is commonly called a Native XML database; for example Xindice an open source database. The advantage to using this database is that data is stored in an XML format. Xpath is used as the query language for the database.

The next type of database is a traditional relational database. While this type of database does not store XML data most commercial relational databases offer XML support. A mapping has to occur between the XML data and the data stored in the relational database. This adds an extra step into inserting and retrieving XML data into the database.

For this project, a relational database is used since military databases are typically relational databases. This decision allows for future research and development to use the system developed in this project with military data sets. Microsoft's SQL 2000 database is used as the relational database in this project. The school has a license for the database and the database provides necessary XML support.

2. Setting Up SQL 2000 for XML Support

The standard version of Microsoft's SQL 2000 server must have patches added to support XML data. SQL 2000 is the current version of Microsoft's SQL database. The specific patch needed for XML support is the SQLXML patch. Once this patch is installed on the server, the server then supports XML. The patch can be found at http://www.microsoft.com/sql/downloads/default.asp (accessed August 2003). Detailed instructions on installation are provided on the site.

3. Testing XML Support on Database

Once the server has had the patch applied, it must be tested to ensure the server is functioning correctly. To do this it is recommended that the SQL Server 2000 Web Services Toolkit from Microsoft be installed. It provides examples that can be run on the server to test the XML output. After the examples are installed and configured correctly, testing can occur.

4. Operational Data Used in This Project

The data set used in this project came from the Digital Chart of the World Server located at Penn State University. The link to the server is http://www.maproom.psu.edu/dcw (accessed August 2003). The Digital Chart of the World (DCW) is an Environmental Systems Research Institute, Inc. (ESRI) product originally developed for the US Defense Mapping Agency (DMA) using DMA data. (Note: DMA is now NIMA). The data used contains latitude and longitude information for the following areas: airfields, cultural, drainage, drainage supplement, land, ocean, and populated places.

5. Conversion of Data for the Project

The data from the site comes in an ArcView interchange file format. This is pulled into Arcview and converted into a Microsoft Excel format. Once in a Microsoft Excel format the data needs to be slightly modified for this project. The modifications consist of deleting data from each table until only the target type, identification, latitude, longitude and elevation remain. Two other rows are added called color and Jpeg. The data is then imported into the database server.

6. Setting Up Tables in the Database

The first pass on setting up tables involves having tables for airfields, cultural, drainage, drainage supplement, land, ocean, and populated places. Development of queries and stored procedures is facilitated by using a single table. One table is created because the data fields in each table are the same. This makes queries on the table easier to work with and is the logical method needed since the data fields are identical. The table has twenty six different target types. The targets include: Airfield, Buildings, Cairn, Camp, CustomsPost, Dam, Fort, GasOilSeparatorPlant, Island, Mine, OilDepot, OilWells, PolicePost, Populated, PowerPlant, PowerStation, PumpingStation, Rocks, Ruins, Substation, Tank, Tomb, Tower, WatchTower, Waterhole, and WaterReservoirs.

7. Stored Procedures and Queries Used in the Database

This system uses eight simple queries, one complex stored procedure and a dynamically produced query using all targets in the 3D scene. The eight simple queries are generated by using Microsoft's Visual Interdev. These queries simply return the content of the eight tables within the database. The stored procedure serves as the main query for the project. See Appendix C for the stored procedure code. It is an involved query that allows the user to select from the twenty six targets found in the main table. It returns the results of the query in an XML file. The last query developed uses a functionality of the SQL 200 server and a node found in VRML. SQL 2000 can have a query run against it from an HTTP tag. VRML has an anchor node that has a uniform resource locator (URL) field. This URL field allows the SQL 2000 server to have query run against it from the 3D scene.

F. DEVELOPMENT OF WEB APPLICATION FOR DATA DISPLAY IN A 3D MODEL

1. Overview and System Architecture

The web interface for this project involves a web site incorporating several servers. The first server is Microsoft's IIS web page server. The second server is the Microsoft's SQL 2000 database server. The last server is a Tomcat server. The Tomcat server is used for Java servlets. The website has a combination of HTML and Active

Server Pages (ASP) pages. Visual Basic (VB) script is used within several pages to perform conversions on latitude and longitude coordinates.

2. Web Site Structure

The layout for the website is relatively simple in design. The purpose of the website is to serve as an interface to the database and the 3D model. ASP pages are used to incorporate the features offered by the tie in between Microsoft's IIS server and the database server.

3. Setting Up the Tomcat Server

The use of a Tomcat server is a key aspect of this project. It allows for the flexibility of using Java servlets and Java classes in the application. Setting up the server involves relatively few changes. Each version of Tomcat can vary slightly with each release. The version used in this project is 4.1.24. Modification is needed to the server.xml file and the web.xml file to enable the root content, invoke the servlet, and turn on servlet reloading.

4. Relationship Between the Database, XML, X3D, VRML and XSLT

The relationship between data in the database and the VRML model is made possible by the use of XML and X3D. As discussed earlier X3D is an XML version of VRML. With the database able to output an XML file a common data format is established between the data and the VRML model. In order to complete the bridge between the data and the VRML code, XSLT is used. XSLT allows the incorporation of the data into a single X3D file. Another XSLT translates the combined X3D file into VRML.

5. XSLT Design

Two XSLT files are used in this project. One was originally designed by Captain James Neushul, USMC, a fellow student, for this application. This XSLT is referred to as the main XSLT. See appendix D for this XSLT file. The second is used in the X3D-Edit application to convert X3D into VRML. The main XSLT developed by Captain Neushul is slightly modified from the initial version. Both XSLT files are executed using the Java servlets on the Tomcat server.

The main XSLT when executed by the Java servlet requires two XML files. The first is the XML file returned by the query on the database. The second is an X3D file providing a prototype for a dome and icon VRML prototype. The main XSLT reads through the query results XML file and identifies the type of target. It then chooses between the icon or dome prototype code. Once the correct object is selected it takes latitude, longitude, and Jpeg information and fills in the fields within the VRML prototype. This continues until all data sets returned from the query are converted into X3D. This process generates a new X3D file. This file is then be converted into VRML by the X3D-Edit XSLT.

6. Development of VRML Prototypes

The development of a VRML prototype in X3D was critical to this project. See Appendix E for the X3D prototype. A VRML prototype allows for a user to re-use a VRML object. This is very similar to how a person uses a Java class in developing a Java application. The VRML prototype can be within the VRML file it is used or can be in another file. The use of VRML prototypes in developing VRML models is extremely useful. For this project, it helped to facilitate the generation of multiple icons and domes to represent the data within the relation database.

7. Java Servlets and Classes Used in Application

There is one Java servlet and three Java classes in this project. The Java servlet is used to allow the use of Java classes within a web application. The servlet is called from the web application and calls two Java classes.

The first Java class called by the Java servlet is called FileFix.class. This Java class was written by Captain Sean Hynes, USMC. The Java class takes in an XML file and strips away spaces found within an XML file. It is used because when the SQL 2000 database server outputted the XML it generates a control return after about 10,000 characters. This causes the resulting XML to be not well formed. The FileFix Java class corrects this problem in the file.

The second Java class called XSLT.class executes the XSLT files. This Java class is executed twice by the Java servlet: the first time is to execute the main XSLT and the second time to execute the X3D-Edit XSLT.

8. Use of VB Script in Web Pages for Latitude and Longitude Conversion

VB script is used on the web page to query the database. The script takes the user's latitude and longitude entry and transforms it into a decimal version of the coordinate. If the user enters 33 22 2, the script converts that into 33.3672222…, the same format used to store coordinates within the database.

G. SUMMARY

This chapter discussed the tools and methods used to research and develop 3D models used to provide the warfighter a 3D perspective of the battlespace. The first section provides an overview of the tools used. The next two sections discuss the development of urban areas and the terrain covering a large area of operation. The last sections cover the database development and the web application used in the exemplar.

IV. DEVELOPMENT RESULTS

A. INTRODUCTION

In this chapter, the assessments made on the research, development and functionality of the urban models, the terrain, database and website is discussed. The chapter follows the same outline as Chapter III. It also discusses the results of the development of the 3D scenes and the interaction with the database in the exemplar.

B. TESTING AND ASSESSMENT

Testing for this project is primarily performed by running the application and ensuring that it works. Observations are made on frame rate, rendering of the designed models, location and placement of objects, appearance of textures, comparison of models with real perspective, and time versus quality in building 3D models.

For the database and web application observations are made in the process used to output XML from the database. Testing on the application consists of ensuring the application functioned as designed. The application has been demonstrated to several audiences over the course of eight months and has run correctly in over twelve different demonstrations.

C. FUNCTIONALITY OF THE 3D MODELS FOR URBAN OPERATIONS

1. Overview

As stated in the beginning of this thesis the actual utility of 3D models in military operations is an opinion of this author. It is an assumption that these models will benefit military operations. This assumption is based upon experience of the author in military operations. The use of 3D models in urban operations provides commanders a perspective of the battlefield that is normally only obtained by putting personnel in harm’s way. For NEOs military commanders may not have the ability to see the area of operation until they are executing the operation.

Having 3D models of American Embassies allows commanders to see the area of operation before they actually arrive at the embassy. This allows commanders to review defense plans, reaction force drill, and review fields of fire.

2. Navigation

As observed in this research, navigation in a 3D model is usually difficult. Trying to move around in an urban 3D model is very tasking on the user. The current controls offered by VRML browsers are inadequate for use in 3D models of urban operations. One navigation mode that is helpful is to use the walk mode. However this does not allow the user to easily move to the top of buildings and can cause other problems. While using the walk mode in an urban scene a user can actually become stuck in the 3D scene. This is caused by gaps in the polygons. An example would be the floor in a room not touching the wall. When the user moves to the wall he could fall between the floor and wall. This can be demonstrated by navigating in the Herrmann Hall model in the NPS campus scene (Figure 42). To compensate for awkward navigation, most authors provide viewpoints that allow users to navigate quickly to key perspectives in the scene. While this approach helps, a user also needs to be able to move freely within the 3D scene. As with the joystick controls developed for use within the Iraq scene, special controls need to be programmed for use in 3D scenes of urban operations.

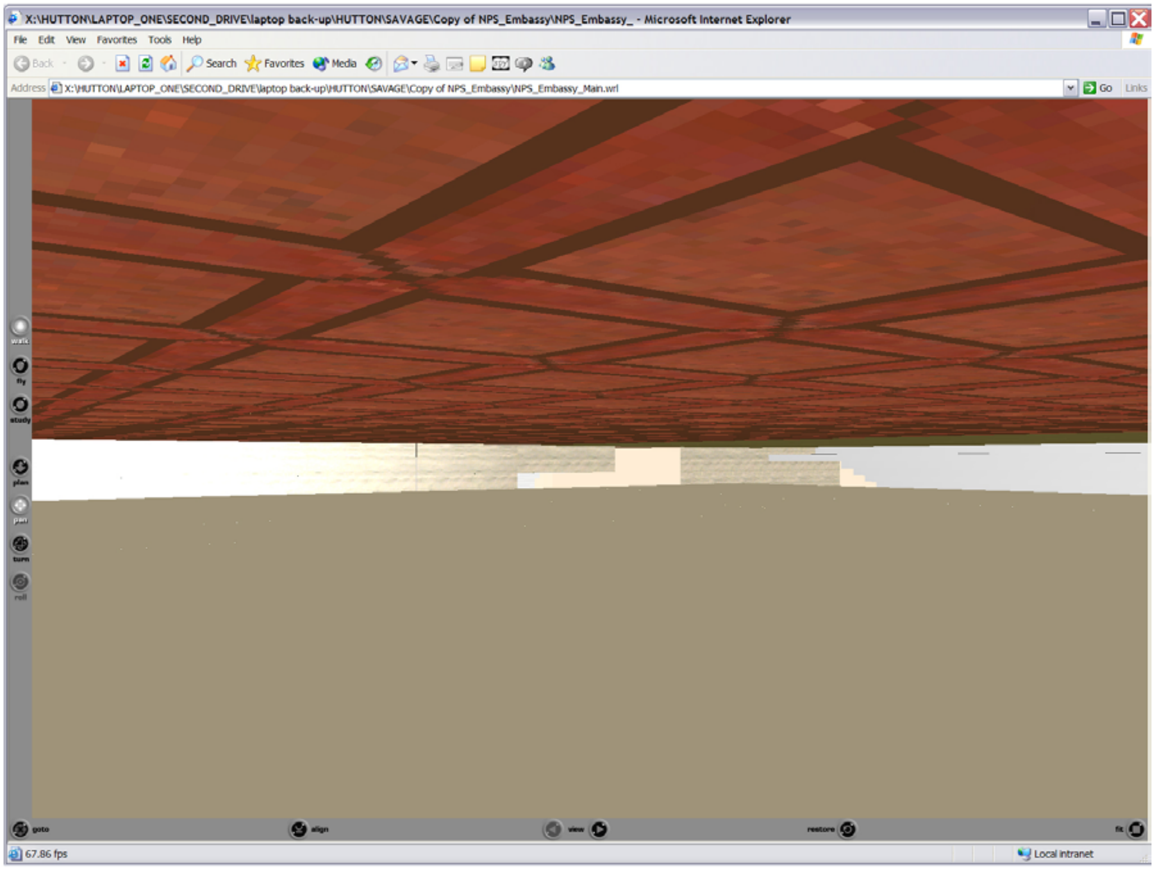

X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\SAVAGE\Copy of NPS_Embassy\NPS_Embassy_ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
X:\HUTTON\LAPTOP_ONE\SECOND_DRIVE\laptop back-up\HUTTON\SAVAGE\Copy of NPS_Embassy\NPS_Embassy_Main.wrl
Go
Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
67.86 fps
Local intranet

Figure 42. Herrmann Hall scene showing user stuck in scene while in walk navigation mode. The floor can be scene overhead.

3. Frame Rate Testing

Frame rate for a VRML 3D scene can be affected by many factors. Some of these factors are related to the user's system and are system dependent. Processor speed, amount of RAM, and the type of graphics card can all affect frame rate. Additionally, the VRML model can affect the frame rate. The polygon count typically has the greatest effect. The higher the polygon count in the visible portion of a scene the lower the frame rate. The use of Java scripts in the 3D scene can also slow the frame rate as seen in the use of the waypoint interpolator. The following table outlines the observed frame rates of the different models developed for this project. A frame rate higher than fifteen frames a second is typically acceptable. As the frame rate drops below fifteen frames per second, movement in the scene becomes slow and choppy.

Scene Name	Frame rate range	Average
NPS Campus With high resolution Herrmann Hall	6–50 frames per second	30 frames per second
Baghdad	.19–18 frames per second	4 frames per second

Table 1. Frame rates for urban scenes.

4. Scaling and Placement of Objects

Scaling of objects can be precise in X3D models. The NPS campus scene demonstrates this capability. While looking at the model from an overhead perspective, it is almost impossible to tell the models of buildings from the image of buildings within the texture. Another technique for checking the scale of objects in a 3D scene is by building a ruler at a set scale. This allows the user to check the actual measurement of an object and ensure the scaling of objects in the scene is accurate.

5. Texture Appearance

The textures used in a 3D model of an urban scene can have a wide range of quality. Since an urban scene is built from different models in a single scene, each texture used in the scene typically only covers a small portion of the scene. Because of this the textures can be smaller and have a lower pixel count without degrading the appearance of the overall model. The textures used in the campus model were mostly smaller than 128 pixels by 128 pixels. The only texture larger than 256 pixels was the image that covered the elevation grid and acted as the foundation for the model. When developing urban models textures should be smaller having a lower pixel count.

6. Comparison of 3D Model with Real Perspective

Several photos were taken on the NPS campus (Figure 43). The view point and perspective that the photos are taken from was then found within the 3D model of the campus. One immediate observation is that the model has no trees in the scene. As seen from the photos and screen shots, the location of the models was reflective of the actual campus.

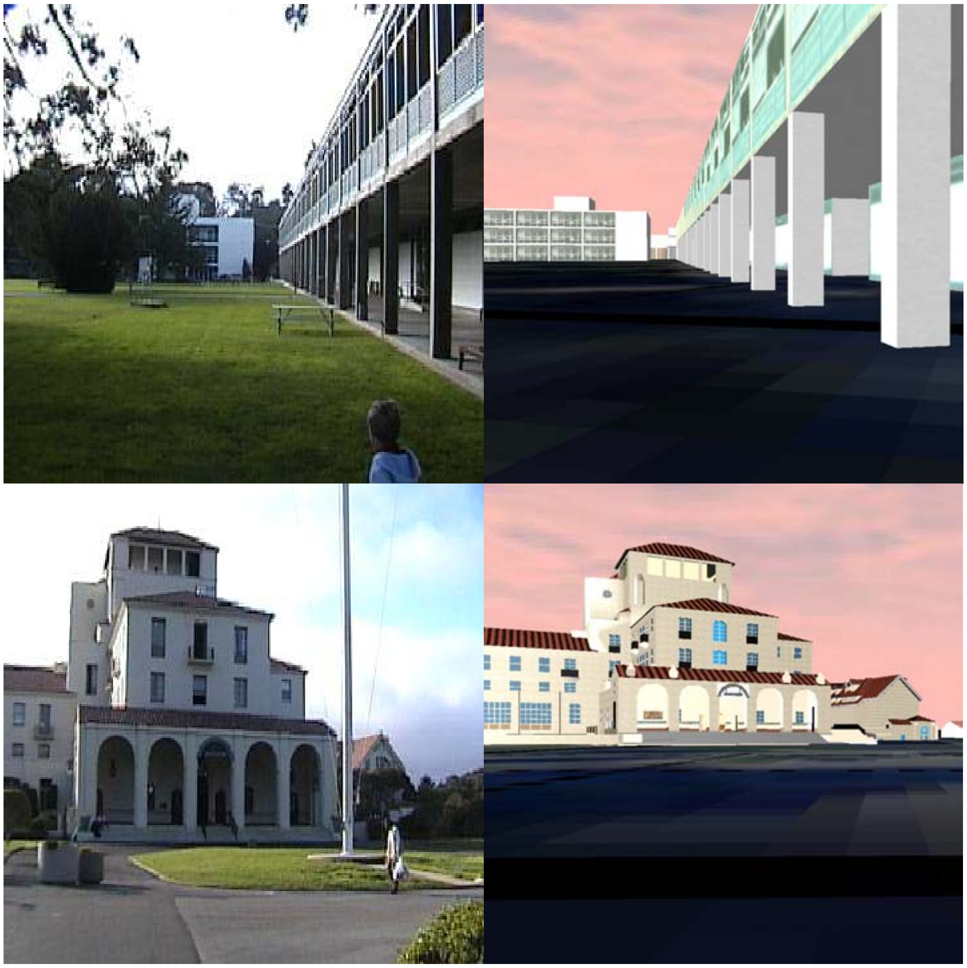


Figure 43. Comparison of campus photos and 3D models.

7. Time Versus Appearance

In developing an urban area, the amount of time spent developing the model is usually indirect proportion to the appearance of the model. The term quality is not used because quality can be measured based upon several factors. A model of the campus that simply reflected the basic dimensions and location of buildings on the campus can be constructed in about eight hours or less. As requirements for a more realistic appearance is required, more time is required. The Herrmann Hall model built by John Locke took an estimated 400 man hours to develop. Rapid development of urban scenes is an important future capability.

D. FUNCTIONALITY OF THE 3D MODEL FOR TERRAIN WITHIN THE AREA OF OPERATION (AO)

1. Overview

The two scenes developed covering Oahu and Iraq demonstrates that a 3D perspective can be presented to a commander covering a defined area of operation. The Iraq model shows a large flat area of operation. The Oahu model shows an area of operation with a smaller island with several large mountain ranges. The end result of both models is a 3D perspective covering a defined area of operation with the actual slope and terrain for the areas.

2. Navigation

The use of the joystick for the Iraq model is extremely useful. It allows users to easily fly over the area of operation and use the map texture to determine the location. The user does not have to depend upon the navigation modes offered by Cortona. While using the joystick, the navigation node is set to NONE to prevent competing controls from interfering. This is because Cortona has built-in support for the use of a joystick. The built-in joystick functionality in Cortona still requires the user to choose the navigation mode and thus only slightly enhances the existing controls.

3. Frame Rate Testing

As discussed in section B of this chapter, the frame rate for a 3D model is dependent upon several factors. The frame rate results of the Oahu and Iraq model are shown in Table 2.

Scene Name	Frame rate range	Average
Oahu	20–40 frames per second	25 frames per second
Iraq	60-90 frames per second	70 frames per second

Table 2. Frame rates for terrain scenes.

4. Scaling and Placement of Objects

Placing objects in the Oahu and Iraq models requires several extra steps. The scaling of the models requires the use of rulers to ensure the correct scale. To ensure the terrain scales correctly for the Oahu model a VRML block measuring 1000 meters along the X Y Z axis is inlined into the scene. The block is transformed to fit one of the 1:50,000 grid squares. This ensures the terrain is correctly scaled. Since the Iraq scene is built on a geocentric model, and the map textures uses latitude and longitude lines, a different method is used to check the scale. To check the scale several 3D X Y Z axes are inserted at several latitude and longitude locations. To check the scale of models in both scenes the same methods described in Section B of this chapter are used.

The location of objects is determined by the X Y Z X3D system for the Oahu model. The Iraq model uses the X3D Geospatial Profile latitude, longitude and elevation coordinates for inserting objects. Inserting objects into both models also requires adjustment of the Y coordinate in the VRML coordinate system due to the contour of the terrain. For inserting objects into the Iraq scene, the Y axis represents elevation. Testing the coordinate system for the Iraq model shows that object locations based upon the latitude, longitude coordinates are relatively accurate. Objects appear within several hundred meters from the location on the map. Variation in object location can be accredited to the stretching of the map texture on the terrain.

5. Map Texture Quality for Terrain

Texture quality in terrain models can be a key aspect of the model. If the texture is an image and is just being used to provide an appealing visual representation, then the quality is up to the developer. If the texture is a map and the location of objects is important, the map texture is important. The size of the texture used in the Iraq scene was 512 pixels by 512 pixels.

One issue with texture use is that textures are stretched. This creates variations in actual locations if there are drastic changes in elevation. Currently the best correction for this is to increase the tiling of textures in the model. Tiling a texture decreases the amount of variation within the model. When building terrain models, the use of the model will determine the required accuracy needed for the application of textures.

6. Time Versus Appearance

The time involved to develop the 3D models of terrain used for this project did not have an impact on the quality of the models. The process used to develop the models is done by using DTED data and a commercial application. The time needed to develop the terrain is measured based upon the amount of terrain being modeled. The Oahu terrain might be constructed in less than one hour. The Iraq scene might take an estimated eight hours because of the tiling technique used. The time estimate for both models does not include the time needed to build a texture for these scenes.

The amount of time to build a 3D model of terrain can take longer depending upon the texture being used. If a texture has to be built, then that time must be added to the process. The textures for the models used in these projects take several hours. This is due to a lack of an application that could read the CDRG CDs from NIMA and output the desired image at the desired size. Currently testing begins with an application that supports reading the CDRG CDs. If this application functions as described building large textures should only take a few minutes. Unfortunately for this author, the texture for Oahu takes over 24 hours to produce. The Iraq textures take about 36 hours. This was partially a learning curve for the first project. Mostly it was just a time-consuming process can be done more quickly with an application designed to work with CDRG maps.

E. RENDERING OF DATA WITHIN A 3D MODEL

1. Design of Database Tables

The design of the database tables for this project is purposefully simple. The tables are designed to support a very simple set of data in order to demonstrate the modeling of data in a 3D model. The key rows in the table are the rows providing information needed to place the objects in the 3D model. The rows needed to place the objects are latitude, longitude, elevation, color, JPEG and target identification rows. Even though these tables are kept simple they demonstrate how relational data can be rendered within a 3D model.

2. Output of XML from Database

One critical aspect of modeling relational data within a 3D model is the ability of the database to output the relational data in an XML format. Most major databases like Microsoft's SQL 2000 and Oracle 9 have this support built into the databases. The purpose of this research is to show that data within a relational database can be used within a 3D scene. This project shows that data can be converted into a 3D model.

One problem encountered in this project was getting the database to write the XML file to a web server. This is not a standard procedure for the database server. Most output resulting from queries on a database server from a web browser are passed back to the browser. For this project, it was desired to have the resulting XML data write to a file. Then the file is modified using Java servlets and XSLT. To get the server to output the file a system stored procedure called spmakewebtask is used. This stored procedure is designed to write files to a designated drive. This allows the XML output to be written to a file.

One problem with this system stored procedure is that it separates the output into 2033 lines of characters. This causes the resulting XML file to be not well formed XML. This happens because of the ODBC it uses does not support a FOR XML clause used in the stored procedure. Microsoft has several solutions in the following article: http://support.microsoft.com/default.aspx?scid=KB;en-us;q275583 (accessed August 2003). For this project a Java class is used to remove the resulting problem. While a relational database is employed in this project, it might be more appropriate to use a native XML database.

3. Use of Java and XSLT

Java and XSLT are used in this project to perform the necessary transformation on the data. A Java servlet performs the multiple XSLT translation. The XSLT enables translation of the relational data into a 3D format. There are no noticeable limitations in the use of these technologies. Furthermore, the flexibility offered by using open source code made this project possible. This is because Java and XSLT enable the programmer to adapt an application. The data translation capability of using XML and XSLT is critical to performing data translation between different data formats.

4. Development of VRML Prototypes

The X3D prototypes developed in this project serves as the 3D objects that model the relational data. These prototypes are an enabling factor in modeling 3D data. One perspective to remember in developing prototypes to model relational data is the polygon count of the 3D objects. If a large data set is modeled in the 3D scene and the prototype has a high polygon count this can result in a slow frame rate. If large data sets are used, the prototypes must be kept simple or other methods involving the use of controls like the X3D Inline node needs to be used.

F. WEB SITE, INTERACTION WITH THE DATABASE, AND RENDERING OF 3D MODELS

1. Layout and Development of Web Site

The web site demonstrates that relational data can be modeled within a 3D scene through a web browser. The site is designed to allow viewing of the relational data in web format and to allow queries to be run against the database. Microsoft IIS and Visual Interdev are extremely helpful in the development of this web site. Visual Interdev has a drag and drop capability to set up the connection and perform queries on the database. There were no developmental issues encountered in the design of the web site or the development of the site.

2. Utilization of Apache Tomcat HTTP Server

The use of Tomcat in this project facilitates the use of XSLT. It allows a Java servlet to be called from a web page in Microsoft's IIS server. This is accomplished by the use of a hyperlink to the Java servlet from a web page on the IIS server. This process avoids having to link the two servers together.

G. SUMMARY

This section covered the results from the development of urban and terrain models used for the research conducted. It has addressed specific issues related to the development of these models in X3D. The last section addressed the interaction with the database and rendering of relational data in the 3D scene.

V. RECOMMENDATIONS AND CONCLUSIONS

A. TECHNICAL FINDINGS

1. Overview

The research for this thesis took over one year. It involved the use of 3D modeling and the tie in of relational data to a 3D scene. Research demonstrates to the warfighter that modeling of operational data can be done within a 3D battlespace. Through the use of a web-based exemplar the modeling of operational data from a relational database into a 3D battlespace is accomplished. Technologies used in this thesis include XML, XSLT, Java, HTML, VB script, and X3D.

2. Visualization of Data

Visualization of data in a 2D format has long been the means by which the warfighter plans and conducts military operations. The visualization of operational data has long been a challenge for military planners. Data used by the military planner is almost always tied to a geographic location. The use of X3D Geospatial Profile in 3D models makes locating data used by the military planner in a 3D battlespace relatively easy.

The research of this thesis serves as a starting point to demonstrate to the warfighter the steps needed to visualize operations data in a 3D format through a web browser. The research performed shows that the use of XML, XSLT, and X3D is the key to accomplishing the visualization of data. The exemplar presented in this research demonstrates that the use of a 3D format can be accomplished.

3. Development of Urban Models

This thesis demonstrates that the development of accurate and realistic 3D models of urban areas can be accomplished. The time spent on the development of 3D models of urban areas is directly related to the appearance of the model. Urban models must be constructed in a manner that ensures users will not get stuck between polygons. This requires close inspection of the model to ensure no gaps between polygons. The use of texture maps can ensure the correct location and scaling of buildings.

4. Development of Terrain

The 3D modeling of terrain covering a given area of operation is demonstrated in this thesis. The 3D modeling of terrain is accomplished using several commercial applications to read and process DTED data. DTED data is readily available to the warfighter. The level of detail desired by the warfighter determines the amount of terrain that can be rendered in the web browser at any one time.

Based upon work accomplished by Captain James Neushul and the research conducted in this thesis it is feasible to develop an application that automatically generates the terrain from the desired level of DTED [Neushul 2003]. The program MultiGen Creator allows the user to select from the area that is desired to be modeled and select options related to the polygon count of the terrain rendered. Additionally, allows areas to be tiled so that the user can use a tiling system when building data sets covering a large area of operation.

While the use of elevation and geo-elevation grids can have a higher level of detail, the area rendered will most likely not cover the entire area needed for most operations. If the terrain is being built for platoon level units and smaller the use of an elevation grid is acceptable for operational planning. In planning for a military operation the commander must be able to see the entire battlespace. The area of operation for a battlespace will typically cover more than 197 square miles or 255 square minutes of latitude and longitude. This is the size of one 1:50,000 map sheet. Earlier observations have shown that if using level two DTED this area cannot be rendered within the web browser. An area approximately 88 square miles using an elevation grid with thirty meter postings is the largest area that can to be rendered. This area uses a 39 kilobyte image and only provides two to three frames per second frame rate.

5. Functionality of Web Application

The use of a web browser for this exemplar demonstrates that relational data can be modeled in a 3D model through a web browser. The ability to use XML, X3D, and XSLT within this web-based application is the key to rendering the data in the 3D model. The flexibility provided by these technologies is clearly demonstrated in this research. Bringing together relational data and 3D models within a web-based application is feasible.

B. BENEFITS

1. Overview

This thesis demonstrates that the modeling of operational data is feasible. It is the assumption and belief of this author that 3D modeling of operational data will benefit military planners. This section outlines the benefits that I believe are possible based upon my experience as a warfighter.

2. Urban Terrain

The 3D modeling of urban areas allows the warfighter to see the area of operation before the forces are committed into the area. Historically urban warfare is known to bring about a higher casualty count than traditional operations. There are many factors that contribute to the difficulty of urban operations. Having a 3D model allows military planners to try to avoid some of the pitfalls of urban areas. Being able to determine what routes tanks can use, where artillery can be effective, and determine fields of fire are just some of the advantages of using a 3D model. Tying this model in with operational data can add even more benefits. Linking intelligence data about buildings and data on hostile forces, identifying restricted targets like religious buildings and hospitals are examples of items that can be modeled. The use of 3D models of urban areas is recommended for the warfigher.

3. Modeling Terrain for an Area of Operation

Providing the commander with a picture of the area of operation is critical to planning for military operations and in control of the battlefield. Having a 3D perspective of the battlefield provides the commander with a range of information that can be obtained by visual observation without a word being spoken to describe the area to the commander. As an intelligence officer the aurthor was once told that the MCOO paints a picture of the battlefield. Modeling of DTED data in a 3D model paints a 3D perspective of the battlefield. It additionally allows the commander to move around the battlespace and see the battlefield from any perspective that he chooses. Merging DTED data with operational data from existing relational databases provides a perspective of the battlespace that this author has not yet seen in any other application in use today by military commanders.

4. Course of Action Development

One benefit of modeling the battlespace in a 3D model is the ability it provides military planners to review courses of action (COA). As time permits, planners will develop several COAs for an operation. The COAs are war gamed between the commander's staff. The commander then chooses a course of action for the operation.

The 3D model allows planners to review the COAs using the modeled 3D battlespace. Seeing how the terrain will affect the COA in a 3D perspective allows for insight that is not easily seen from a 2D map. Determining how mountains will affect communications, seeing if routes over terrain can slow movement can all be modeled in the 3D battlespace. These are just some of the time consuming tasks that a commander's staff must develop for a COA review. By having a 3D model set up for COA planning these tasks can easily be automatically completed by a 3D model.

5. Consolidating, Reviewing and Analyzing Operation Data

Because the 3D models used in this thesis use X3D being able to pull in operational data from several different data sets is possible. This allows military planners to easily pull in data from different stove-piped systems that originated in different data formats. A common schema for the data sets needs to be developed. Currently work is being conducted on the Generic Hub [NATO 2002] that can be used as the common language for these different data sets.

6. Open Source Development

The fact that the X3D, XML, XSLT and Java are all open standards allows for easy modification of any application used or developed for use by the military that is based on these technologies. Modifications and integration of current stove-piped systems are hindered because most of these systems use proprietary coding. Moving applications into a web-based arena will provide the warfighter with a greater level of flexibility. It will allow for quick and easy modification of the developed applications.

C. RECOMMENDATIONS

1. Overview

From the research of this thesis and the development of an exemplar application, it is recommended that research be continued in the development of a 3D battlespace for the warfighter. The technologies used in thesis can be used to develop tools that will present operational data within a 3D battlespace. Work can be started to begin model actual military data sets currently used by military planners. Modeling of data from national agencies like the National Security Agency (NSA) and Defense Intelligence Agency (DIA) can be started. The structure of the data is all that is needed to begin the process. Modeling of actual classified data is a final phase. The developed models can cover large areas of operations and pull in different data sets. Expanding upon the concepts in this thesis will quickly bring about a fully functional web-based application to provide the commander with a force multiplier in planning and conducting military operations.

2. Future Development

a. Conversion of Entire DTED Library

The DTED terrain data seldom changes. Major changes in terrain occur over time and are easily noticed. Because of this it is recommended that the entire DTED set covering the world be converted into an X3D format. This process can be started with level 1 DTED data and modeled using indexed face sets. This research can be conducted using DTED data from an area with drastic changes in elevation to determine the largest possible size for an X3D terrain tile file. The files can be constructed based upon the WGS84 ellipsoid model used by NIMA to produce military maps. The process can be automated so that any changes can easily generate updates for selected tiles. It is recommended that be done in close coordination with NIMA.

b. Scalable Vector Graphics (SVG)

One limitation encountered in using VRML is that it does not support the use of Scalable Vector Graphics (SVG). SVG formatted images change in scale based upon the proximity to the image. For use of military maps or images in 3D models of an area of operation this is critical. Having a road become blurred because of pixel replication when the viewpoint gets close to the image can cause problems with navigation and appearance in the model. It is strongly recommended that the X3D specification include the ability for the models to use SVG textures.

c. Continued Development with X3D Geospatial Profile

The use of X3D Geospatial Profile in military applications is critical to modeling operational data. Without the latitude and longitude coordinate systems and military grid reference system in the model placement of data sets becomes almost impossible. Work the X3D Geospatial Profile functionality should be continued.

d. Navigation Aids

The use of the VRML devicehandler provides easy navigation within the 3D VRML scenes. It is recommended that this functionality be developed for X3D. Additionally, prototypes that output navigation information must be developed. These prototypes will allow the developer to enable the user to know his location within the 3D battlespace at all times.

e. Prototype Development and MilStd 2525b

The development of a standard format for displaying operational data in a 3D scene must be researched. The standard will allow for the display of both a low and high polygon count representation of the data. The current military standard developed by the Defense Information Systems Agency (DISA) is designed for symbols used in a 2D perspective [DEFENSE 1999]. When developing 3D symbols, this standard must be conformed to as closely as possible. Any exceptions that may be needed because of the 3D battlespace must be noted and reported to DISA.

f. Near-Real-Time Control of Remote Targeting Via Network Protocols

Research into XML protocols has been conducted by Ekrem Serin-Lieutenant Junior Grade, Turkish Navy. His thesis can be referenced at http://theses.nps.navy.mil/03Mar_Serin.pdf (accessed August 2003). Continued work in this area needs to be pursued to allow commanders to update the battlespace with near-real time data. This will allow the commander to observe combat operations as they are being conducted.

g. Conversion of Web-based Application to Web Service Application

Development of a web services based application that can access multiple data sets is the next logical step for the exemplar developed in this research. This will aid in the development of a service to allow the warfighter to access and combine multiple data sets in a 3D battlespace. The service must have the ability to provide both static data from databases and near-real time data from a network feed.

3. Scalable 3D Combined Operational Picture (3D COP) of the Battlefield

The ground work for a 3D COP is established. The Extensible Modeling and Simulation Framework (XMSF) project started at NPS is defining concepts for the next generation for modeling and simulation [XMSF 2003]. The concepts outlined in the XMSF can also be applied towards the development of a 3D COP. Current modeling and simulation structure used in training Joint Task Forces (JTF) typically integrates service specific command and control systems to provide a COP of the battlefield. While the development of a COP has been demonstrated the next generation COP needs to be a 3D COP of the battlefield.

a. Phase One: Replicating the Current 2D COP in a 3D COP Battlespace

The first phase has already begun. In an exercise with J9 from Joint Forces Command (JFCOM), a limited view of the 3D battlespace was demonstrated. A feed from the model and simulation structure was transformed into a 3D battlespace. This concept needs to be refined and development needs to continue. Terrain covering the entire area of operation needs to be built. This involves building a X3D terrain library from DTED data. The next step is to continue development of the icons used to represent the friendly, enemy civilian and other targets on the battlespace. One important factor in the development of the icons is the splitting–out and aggregation of military units as the user zooms in an out of the battlespace. This phase is simply meant to augment the current 2D COP used in command and control systems on the battlefield today. The three key items to complete in this phase are the development of the terrain, icons and the replication of the current COP from a network feed.

b. Phase Two: Exposing Non-traditional Data Sources Via a Web Services Architecture

The command and control systems on the battlefield today are still limited by the data specific stove-piped systems used by the different services and agencies today. Non-traditional data sets like human intelligence reports are still not reflected in the 2D COP. These reports typically are in the form of emails and military messages. Because of the extensibility of the web-based and XML-based XMSF integrating non-traditional report formats can be accomplished. The completion of this phase will outline the steps needed to integrate different report formats not provided by the 2D COP and successfully integrate several commonly used reports. Additionally, the user will begin to see tools that allows the user to select what data to display.

c. Phase Three: Leverages the Integration of Artificial Intelligence (AI) and Agent Based Tools

This final phase enables the user to use AI, agent based tools and physics based 3D modeling to aid in analysis of the battle space. Professor John Hiles from NPS describes the use of agent based modeling to model the actions of terrorist in an asymmetric battlespace reacting to information obtained from different intelligence sources. [Hiles 2003] This is the type of tool needed to filter the vast amounts of intelligence that is collected on the battlefield today. AI and agent based tools combined with the ability of the 3D battlespace to allow physics to be applied in the model provide a force multiplier for our warfighters. There is no true completion to the final phase of this concept. Because of the extensible nature in which it is developed it will continue to develop along with emerging technology. This is the true nature of the strengths offered by development in an open source and extensible framework.

4. Conclusion

This thesis has demonstrated that a 3D perspective of the battlespace can be provided to the warfighter. It demonstrated how both urban and large areas of operation can be modeled in a 3D scene. It demonstrated how relational data can be displayed within a 3D scene representing the battlespace. In an article recently published retired Vice Admiral Arthur K. Cebrowski was quoted as saying “If you are not on a net, you’re not benefiting. … You’re not a part of the Information Age. You’re gone.” [Onley 2003]. His comments were about military operations related to Operation Iraqi Freedom. He outlined issues related to military intelligence and the integration and analysis of data collected on the battlespace. The exemplar demonstrated in this thesis might be an example of a how to address the issues he outlined and provide a tool to aid the warfighter on today’s battlefield.

APPENDIX A. ACRONYMNS AND ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
AT/FP	Anti-Terrorism / Force Protection
CDRG	Compressed ARC Digitized Raster Graphics
CLB	Coastal Landing Beaches
COA	Course of Action
COP	Common Operational Picture
DCW	Digital Chart of the World
DED	Digital Elevation Data
DIA	Defense Intelligence Agency
DISA	Defense Information Systems Agency
DMA	US Defense Mapping Agency
DTED	Digital Terrain Elevation Data
DTD	Document Type Definition
DoD	Department of Defense
GCCS	Global Command and Control System
GIS	Geographic Information System
HLZ	Helicopter Landing Zones
HTML	Hypertext Markup Language
IAS	Intelligence Analysis System
JAVA	Programming language developed by Sun Microsystems
JPEG	Joint Photographic Entertainment Group format for image files
JTF	Joint Task Forces

M&S	Modeling and Simulation
MCOO	Military Combined Obstacle Overlay
NEO	Non-Combatant Evacuation Operation
NIMA	National Imagery and Mapping Agency
NATO	North Atlantic Treaty Organization
NPS	Naval Postgraduate School
NSA	National Security Agency
OPORDER	Operations Order
SAVAGE	Scenario Authoring and Visualization for Advanced Graphical
SGML	Standard Generalized Markup Language
SVG	Scalable Vector Graphics
TBMCS	Theater Battle Management Control System
URL	Uniform Resource Locator
USMTF	United States Message Text Format
UTM	Universal Transverse Mercator
X3D	Extensible 3D Graphics
XMSF	Extensible Modeling and Simulation Framework
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language for Transformation

APPENDIX B. X3D GEOSPATIAL PROFILE CODE USED IN TERRAIN TILES

The code in this appendix was the code found at the top of each of the 25 terrain VRML files used for this project. The code displays the structure for each of the X3D Geospatial prototypes used to enable the terrain to use a latitude and longitude geo-reference system. This provides backwards capability from X3D to VRML 97 browsers.

```
#VRML V2.0 utf8
# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl
# http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl

# [X3D] VRML V3.0 utf8
EXTERNPROTO GeoCoordinate [
  field  SFNode    geoOrigin    # NULL
  field  MFString  geoSystem    # [ "GDC" ]
  field  MFString  point        # []
] [
    "GeoVRML/1.1/protos/GeoCoordinate.wrl#GeoCoordinate"
    "C:/Program Files/GeoVRML/1.1/protos/GeoCoordinate.wrl#GeoCoordinate"
    "file:///C|/Program Files/GeoVRML/1.1/protos/GeoCoordinate.wrl#GeoCoordinate"
    "urn:web3d:geovrml:1.1/protos/GeoCoordinate.wrl#GeoCoordinate"
    "http://www.geovrml.org/1.1/protos/GeoCoordinate.wrl#GeoCoordinate"
  ]
EXTERNPROTO GeoElevationGrid [
  field        SFNode    geoOrigin        # NULL
  field        MFString  geoSystem        # [ "GDC" ]
  field        SFString  geoGridOrigin    # "0 0 0"
  field        SFInt32   xDimension       # 0
  field        SFString  xSpacing         # "1.0"
  field        SFInt32   zDimension       # 0
  field        SFString  zSpacing         # "1.0"
  field        SFFloat   yScale           # 1.0
  field        MFFloat   height           # []
  eventIn      SFFloat   set_yScale
  eventIn      MFFloat   set_height
  exposedField  SFNode   color            # NULL
  exposedField  SFNode   texCoord         # NULL
  exposedField  SFNode   normal           # NULL
  field        SFBool    normalPerVertex   # TRUE
  field        SFBool    ccw              # TRUE
  field        SFBool    colorPerVertex   # TRUE
  field        SFFloat   creaseAngle      # 0
  field        SFBool    solid            # TRUE
] [
    "GeoVRML/1.1/protos/GeoElevationGrid.wrl#GeoElevationGrid"
    "C:/Program Files/GeoVRML/1.1/protos/GeoElevationGrid.wrl#GeoElevationGrid"
    "file:///C|/Program Files/GeoVRML/1.1/protos/GeoElevationGrid.wrl#GeoElevationGrid"
    "urn:web3d:geovrml:1.1/protos/GeoElevationGrid.wrl#GeoElevationGrid"
    "http://www.geovrml.org/1.1/protos/GeoElevationGrid.wrl#GeoElevationGrid"
  ]
EXTERNPROTO GeoMetadata [
  exposedField MFString url        # []
  exposedField MFString summary    # []
  exposedField MFNode   data       # []
] [
    "GeoVRML/1.1/protos/GeoMetadata.wrl#GeoMetadata"
    "C:/Program Files/GeoVRML/1.1/protos/GeoMetadata.wrl#GeoMetadata"
    "file:///C|/Program Files/GeoVRML/1.1/protos/GeoMetadata.wrl#GeoMetadata"
    "urn:web3d:geovrml:1.1/protos/GeoMetadata.wrl#GeoMetadata"
    "http://www.geovrml.org/1.1/protos/GeoMetadata.wrl#GeoMetadata"
  ]
EXTERNPROTO GeoOrigin [
 exposedField  MFString  geoSystem    # [ "GDC" ]
 exposedField  SFString  geoCoords    # ""
 field         SFBool    rotateYUp    # FALSE
] [
    "GeoVRML/1.1/protos/GeoOrigin.wrl#GeoOrigin"
    "C:/Program Files/GeoVRML/1.1/protos/GeoOrigin.wrl#GeoOrigin"
    "file:///C|/Program Files/GeoVRML/1.1/protos/GeoOrigin.wrl#GeoOrigin"
    "urn:web3d:geovrml:1.1/protos/GeoOrigin.wrl#GeoOrigin"
    "http://www.geovrml.org/1.1/protos/GeoOrigin.wrl#GeoOrigin"
  ]
  EXTERNPROTO InlineLoadControl [
  field        SFVec3fbboxCenter
  exposedField MFString        url
  eventOut     MFNode          children
  field        SFVec3fbboxSize
  exposedField SFBool          load
][
  "urn:web3d.org:vrml97:node:InlineLoadControl"
"file:///C:/Program%20Files/GeoVRML/1.1/protos/InlineLoadControl.wrl"
"http://www.geovrml.org/1.1/protos/InlineLoadControl.wrl"
 ]
 EXTERNPROTO GeoLocation [
   exposedField SFString  geoCoords
   field        MFNode    children
   field        SFNode    geoOrigin
   field        MFString  geoSystem
] [ "urn:web3d.org:vrml97:node:GeoLocation"
    "file:///C:/Program%20Files/GeoVRML/1.1/protos/GeoLocation.wrl"
    "http://www.geovrml.org/1.1/protos/GeoLocation.wrl" ]

# [Scene]
```

APPENDIX C. X3D CODE FOR FLIGHT CONTROLS USD TO CONTROL V22 IN IRAQ SCENE

The appendix is the code that uses the devicehandler to control the V22 aircraft in the Iraq scene using a joystick. The code was originally developed by Eric Maranne. The code is modified from the original example and is now used simulate controls of an aircraft. The prototype allows the user to use external device and map actions from that device to control actions in X3D scene.

```
#VRML V2.0 utf8
# X3D-to-VRML-97 XSL translation autogenerated by X3dToVrml97.xsl
# http://www.web3D.org/TaskGroups/x3d/translation/X3dToVrml97.xsl
# [X3D] VRML V3.0 utf8
# [head]
# Import note: the following meta tags were created during Vrml97ToX3d translation. Please update or delete them as appropriate.
# [meta] filename: FLIGHT.x3d
# [meta] description: Flight Controls for the V22
# [meta] author: Eric Maranne
# [meta] translator: Claude Hutton
# [meta] imported: 17 February 2003
# [meta] revised: 17 February 2003
# [meta] generator: X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-Edit.html
# [meta] generator: Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html
# [Scene]
EXTERNPROTO EMIExtDevHandler [
  eventOut   SFBool        LOOK_DOWN
  eventOut   SFBool        Action_19
  eventOut   SFBool        Action_18
  eventOut   SFBool        Action_17
  eventOut   SFBool        TRACING
  eventOut   SFBool        Action_16
  eventOut   SFBool        Action_15
  eventOut   SFBool        PAN_UP
  eventOut   SFBool        Action_14
  eventOut   SFBool        Action_13
  eventOut   SFBool        Action_12
  eventOut   SFBool        Action_11
eventOut    SFBool      Action_10
eventOut    SFBool      PAN_LEFT
eventOut    SFBool      DISPLAY_MENU
eventOut    SFBool      QUIT
eventOut    SFInt32     PAN_LEFT_RIGHT_AXIS
field     SFBool  pollAtStartUp
eventOut    SFInt32     POINTER_2
eventOut    SFInt32     POINTER_1
eventOut    SFBool      WALK_FLY_TOGGLE
eventOut    SFInt32     POV
eventOut    SFBool      LOOK_LEFT
eventOut    SFInt32     UP_DOWN_AXIS
eventIn    SFBool       poll
eventOut    SFBool      ADD_OBJECT
eventOut    SFInt32     THRUST_AXIS
eventOut    SFBool      Action_50
eventOut    SFBool      LOOK_RAZ
eventOut    SFBool      Action_49
eventOut    SFInt32     PAN
eventOut    SFBool      Action_48
eventOut    SFBool      Action_47
eventOut    SFBool      Action_46
eventIn    SFTime       define
eventOut    SFBool      Action_45
eventOut    SFBool      Action_44
eventOut    SFBool      Action_43
eventOut    SFBool      Action_42
eventOut    SFBool      Action_41
eventOut    SFBool      Action_40
eventOut    SFInt32     ROTATEOBJECT_LEFT_RIGHT_AXIS
field     SFString settingsName
eventOut    SFInt32     ROTATEOBJECT_UP_DOWN_AXIS
eventOut    SFBool      LOOK_UP
eventOut    SFBool      Action_9
eventOut    SFBool      Action_8
eventOut    SFBool      LOOK_RIGHT
eventOut    SFBool      Action_7
eventOut    SFBool      Action_39
eventOut    SFBool      Action_6
eventOut    SFBool      Action_38
```

```
  eventOut    SFInt32      PAN_UP_DOWN_AXIS
  eventOut    SFBool       Action_5
  eventOut    SFBool       Action_37
  eventOut    SFBool       Action_4
  eventOut    SFBool       Action_36
  eventOut    SFBool       Action_3
  eventOut    SFBool       Action_2
  eventOut    SFBool       Action_35
  eventOut    SFBool       Action_1
  eventOut    SFBool       Action_34
  eventOut    SFBool       Action_0
  eventOut    SFBool       Action_33
  eventOut    SFBool       Action_32
  eventOut    MFInt32      OBJ_2_ROT
  eventOut    SFBool       Action_31
  eventOut    SFBool       Action_30
  eventOut    SFBool       ACTIVATION
  eventOut    SFInt32      BANK_AXIS
  eventOut    SFBool       PAN_DOWN
  eventOut    SFBool       Action_29
  eventOut    SFBool       Action_28
  eventOut    SFBool       Action_27
  eventIn    SFTime        timeStep
  eventOut    SFBool       Action_26
  eventOut    SFBool       Action_25
  eventOut    SFBool       Action_24
  eventOut    SFBool       Action_23
  eventOut    SFBool       Action_22
  eventOut    SFBool       PAN_RIGHT
  eventOut    SFBool       Action_21
  eventOut    SFBool       Action_20
  eventOut    SFInt32      LEFT_RIGHT_AXIS
  eventOut    MFInt32      OBJ_1_ROT
  eventOut    SFBool       PICK_OBJECT
][
  "EMIExtDevHandler.wrl"
 ]
NavigationInfo {
  type [ "NONE" ]
}
```

```
PROTO YRotZPosToVP [
 eventIn     SFBool          up
 eventIn     SFInt32         aYAxis
 eventIn     SFInt32         POV
 eventIn     SFInt32         aThrustAxis
 eventIn     SFInt32         aXAxis
 eventIn     SFTime          upd
 eventIn     SFBool          down
 eventIn     SFInt32         aZAxis
 field       MFNodeobject [
  DEF V22 Transform {
   children [
     DEF WitchFol_16 Transform {
      children [
        Viewpoint {
         description "FollowV22"
         orientation 0.0 1.0 0.0 0.0
         position -38.0 10.0 50.0
        }
      ]
     }
     DEF WitchPOV_17 Transform {
      children [
        Viewpoint {
         description "SeeV22View"
         orientation 0.0 1.0 0.0 0.0
         position 8.0 8.0 -4.0
        }
      ]
     }
     Transform {
      rotation 0.0 1.0 0.0 1.57
      children [
        Inline {
         url [ "models/V22/V22.wrl"
]
        }
### Hint:  For maximum portability, append alternate "http://address" to url='"V22.wrl"'
      ]
     }
```

```
    ]
  }
]
]
{
 DEF userVPTrans Transform {
  children [
    DEF PovTransform Transform {
           children IS object
    }
  ]
 }
 DEF Tmr TimeSensor {
  loop TRUE
 }
 DEF S Script {
  field      SFFloat          aThrust 10.0
  eventIn    SFInt32          aThrustAxis IS aThrustAxis
  eventIn    SFTime                   upd IS upd
  eventIn    SFInt32          POV IS POV
  field      SFNode    aTransform     USE userVPTrans
  field      SFVec3f          curPos 0.0 0.0 0.0
  field      SFBool           flagup FALSE
  eventIn    SFBool                   up IS up
  eventIn    SFInt32          aXAxis IS aXAxis
        eventIn    SFInt32 aYAxis IS aYAxis
  eventIn    SFInt32          aZAxis IS aZAxis
  field      SFNode    PovTransform     USE PovTransform
        field      SFRotation         XcurRot 1.0 0.0 0.0 0.0
        field      SFRotation         ZcurRot 0.0 0.0 1.0 0.0
  field      SFRotation       YcurRot  0.0 1.0 0.0 0.0
  field      SFBool           flagdown FALSE
  eventIn    SFBool                   down IS down
  directOutput TRUE
  mustEvaluate TRUE
        url [ "javascript:
                function aThrustAxis (val)
                {
                        aThrust = (100 - val)/ 20 ;
                        curPos.z = (Math.abs(val)  > 5) ? (  (val ) *  aThrust ) : 0;// * 3.14;
```

```
            //      aThrust = (100 - val)/ 100;
                    //print(aThrust);
                    //print(val);
                    //val = (-101 - val);
                    //print(val);
                    //curPos.z = (  (val ) *  aThrust ) ;// * 3.14;

            }
            function aXAxis (val)
            {
                    XcurRot.angle =  (Math.abs(val)  > 5) ? ((  val / 1000) * (aThrust / 20)) : 0;
            }
            function aYAxis (val)
            {
                     YcurRot.angle =  (Math.abs(val)  > 5) ? ((  -val / 1000) * (aThrust / 20)) : 0;
            }
            function aZAxis (val)
            {
                     ZcurRot.angle =  (Math.abs(val)  > 5) ? ((  -val / 1000) * (aThrust / 20)) : 0;
            }
            function POV (val) {

                    switch (val)
                    {
                            case 0  :   {PovTransform.rotation = new SFRotation(-1, 0, 0, .72);
break;}
                    case 4500 :         {PovTransform.rotation = (new SFRotation(-0.683318, -
0.683318, -0.25028, 1.006985)); break;}
                    case 9000 :         {PovTransform.rotation =  (new SFRotation(0, -1, 0, .72));
break;}
                    case 13500 : {PovTransform.rotation =  (new SFRotation( 0.683318, -
0.683318,0.25028, 1.006985)); break;}
                    case 18000 : {PovTransform.rotation =  (new SFRotation(1, 0, 0, .72)); break;}
                    case 22500 : {PovTransform.rotation =  (new SFRotation(0.683318,0.683318,-
0.25028, 1.006985)); break;}
                    case 27000 : {PovTransform.rotation = (new SFRotation(0, 1, 0, .72)); break;}
                    case 31500 : {PovTransform.rotation =  (new SFRotation(-0.683318, 0.683318,
0.25028, 1.006985)); break;}
                    default : {PovTransform.rotation  = new SFRotation(0, 1, 0, 0); break;}
                    }
            }
            function up (val)
```

```
                        {
                                curPos.y = val ? (aThrust * 12) : 0 ;
                        }
                        function down (val)
                        {
                                curPos.y = -val ? (-aThrust * 12) : 0 ;
                        }
                        function upd ()
                        {

        //                      aTransform.center= curPos;
                                aTransform.rotation = aTransform.rotation.multiply (YcurRot);
                                aTransform.rotation = aTransform.rotation.multiply (XcurRot);
                                aTransform.rotation = aTransform.rotation.multiply (ZcurRot);
                                //print(aTransform.translation);
                                aTransform.translation = aTransform.translation.add
(aTransform.rotation.multVec(curPos));
        //                      if (flagup) aTransform.translation = aTransform.translation.add
(aTransform.rotation.multVec(new SFVec3f (0,aThrust,0)));
        //                      if (flagdown) aTransform.translation = aTransform.translation.add
(aTransform.rotation.multVec(new SFVec3f (0,-aThrust,0)));
                        }

        " ]
          }
          ROUTE Tmr.time TO S.upd
        }
        DEF Devices EMIExtDevHandler {
        }
        DEF AXIS YRotZPosToVP {
        }
        ROUTE Devices.LEFT_RIGHT_AXIS TO AXIS.aYAxis
        ROUTE Devices.UP_DOWN_AXIS TO AXIS.aXAxis
        ROUTE Devices.BANK_AXIS TO AXIS.aZAxis
        ROUTE Devices.THRUST_AXIS TO AXIS.aThrustAxis
        ROUTE Devices.POV TO AXIS.POV
        ROUTE Devices.Action_0 TO AXIS.up
        ROUTE Devices.Action_1 TO AXIS.down
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. MAIN STORED PROCEDURE USED IN PROJECT

This appendix is the stored procedure used to conduct the query on the database performed by the user. The stored procedure takes in set of parameters from the user and then initiates a query on the database. When the query is returned it calls a system stored procedure that outputs the results of the query in an XML document. The file is named newtestxml.sql.

```
CREATE PROCEDURE newtestxml
@Airfield varchar(255) = NULL,
@Buildings varchar(255) = NULL,
@Cairn varchar(255) = NULL,
@Camp varchar(255) = NULL,
@CustomsPost varchar(255) = NULL,
@Dam varchar(255) = NULL,
@Fort varchar(255) = NULL,
@GasOilSeparatorPlant varchar(255) = NULL,
@Island varchar(255) = NULL,
@Mine varchar(255) = NULL,
@OilDepot varchar(255) = NULL,
@OilWells varchar(255) = NULL,
@PolicePost varchar(255) = NULL,
@Populated varchar(255) = NULL,
@PowerPlant varchar(255) = NULL,
@PowerStation varchar(255) = NULL,
@PumpingStation varchar(255) = NULL,
@Rocks varchar(255) = NULL,
@Ruins varchar(255) = NULL,
@Substation varchar(255) = NULL,
@Tank varchar(255) = NULL,
@Tomb varchar(255) = NULL,
@Tower varchar(255) = NULL,
@WatchTower varchar(255) = NULL,
@Waterhole varchar(255) = NULL,
@WaterReservoirs varchar(255) = NULL,
@minlat nvarchar(255) = NULL,
@maxlat nvarchar(255) = NULL,
@minlong nvarchar(255) = NULL,
@maxlong nvarchar(255) = NULL

AS

DECLARE  @querystr varchar(8000),
@nAirfield varchar(255),
@nBuildings varchar(255),
@nCairn varchar(255),
@nCamp varchar(255),
@nCustomsPost varchar(255),
@nDam varchar(255),
@nFort varchar(255),
@nGasOilSeparatorPlant varchar(255),
@nIsland varchar(255),
@nMine varchar(255),
@nOilDepot varchar(255),
@nOilWells varchar(255),
@nPolicePost varchar(255),
@nPopulated varchar(255),
@nPowerPlant varchar(255),
@nPowerStation varchar(255),
@nPumpingStation varchar(255),
@nRocks varchar(255),
@nRuins varchar(255),
@nSubstation varchar(255),
@nTank varchar(255),
@nTomb varchar(255),
@nTower varchar(255),
@nWatchTower varchar(255),
@nWaterhole varchar(255),
@nWaterReservoirs varchar(255),
@nminlat nvarchar(255),
@nmaxlat nvarchar(255),
@nminlong nvarchar(255),
@nmaxlong nvarchar(255)

SET @nAirfield = @Airfield
SET @nBuildings = @Buildings
SET @nCairn = @Cairn
SET @nCamp = @Camp
SET @nCustomsPost = @CustomsPost
SET @nDam = @Dam
SET @nFort = @Fort
SET @nGasOilSeparatorPlant = @GasOilSeparatorPlant
SET @nIsland = @Island
SET @nMine = @Mine
SET @nOilDepot = @OilDepot
SET @nOilWells = @OilWells
SET @nPolicePost = @PolicePost
SET @nPopulated = @Populated
SET @nPowerPlant = @PowerPlant
SET @nPowerStation = @PowerStation
SET @nPumpingStation = @PumpingStation
SET @nRocks = @Rocks
SET @nRuins = @Ruins
SET @nSubstation = @Substation
SET @nTank = @Tank
SET @nTomb = @Tomb
SET @nTower = @Tower
SET @nWatchTower = @WatchTower
SET @nWaterhole = @Waterhole
SET @nWaterReservoirs = @WaterReservoirs
SET @nminlat = @minlat
SET @nmaxlat = @maxlat
SET @nminlong = @minlong
SET @nmaxlong = @maxlong

SELECT @querystr =
'
 SELECT *
 FROM IRAQ
 WHERE (
(
(TARGET_TYPE = ''' + @nAirfield + ''') OR
(TARGET_TYPE = ''' + @nBuildings + ''') OR
(TARGET_TYPE = ''' + @nCairn + ''') OR
(TARGET_TYPE = ''' + @nCamp + ''') OR
(TARGET_TYPE = ''' + @nCustomsPost + ''') OR
(TARGET_TYPE = ''' + @nDam + ''') OR
```

```
    (TARGET_TYPE = ''' + @nFort + ''') OR
    (TARGET_TYPE = ''' + @nGasOilSeparatorPlant + ''') OR
    (TARGET_TYPE = ''' + @nIsland + ''') OR
    (TARGET_TYPE = ''' + @nMine + ''') OR
    (TARGET_TYPE = ''' + @nOilDepot + ''') OR
    (TARGET_TYPE = ''' + @nOilWells + ''') OR
    (TARGET_TYPE = ''' + @nPolicePost + ''') OR
    (TARGET_TYPE = ''' + @nPopulated + ''') OR
    (TARGET_TYPE = ''' + @nPowerPlant + ''') OR
    (TARGET_TYPE = ''' + @nPowerStation + ''') OR
    (TARGET_TYPE = ''' + @nPumpingStation + ''') OR
    (TARGET_TYPE = ''' + @nRocks + ''') OR
    (TARGET_TYPE = ''' + @nRuins + ''') OR
    (TARGET_TYPE = ''' + @nSubstation + ''') OR
    (TARGET_TYPE = ''' + @nTank + ''') OR
    (TARGET_TYPE = ''' + @nTomb + ''') OR
    (TARGET_TYPE = ''' + @nTower + ''') OR
    (TARGET_TYPE = ''' + @nWatchTower + ''') OR
    (TARGET_TYPE = ''' + @nWaterhole + ''') OR
    (TARGET_TYPE = ''' + @nWaterReservoirs + ''')
    )
     And  ((LAT)>=''' + @nminlat + ''' And (LAT)<= ''' + @nmaxlat + ''') AND ((LONG)>=''' + @nminlong + ''' And (LONG)<=''' + @nmaxlong + ''')
    )
     FOR XML AUTO

    '
    set nocount on
    EXEC sp_makewebtask
            @outputfile = 'C:\Inetpub\wwwroot\BDV\Temp\queryresults.xml',
              @query = @querystr
    ,


              @templatefile ='C:\Inetpub\wwwroot\BDV\Temp\output.txt'
    GO
```

APPENDIX E. XSLT TO CONVERT XML QUERY RESULTS FILE INTO X3D FORMAT

This XSLT takes an X3D prototype file and the XML query results file and generates the X3D file that is the 3D representation of the relation data.

```
<?xml version="1.0"?>
<xsl:stylesheet   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"   xmlns:xslt="http://xml.apache.org/xslt"
version="1.0">
    <xsl:output method="xml" encoding="utf-8"/>

    <xsl:param name="Protos" select="document('iconDomeProto.x3d')"/>
    <xsl:param name="Data" select="document('newqueryresults.xml')"/>
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <xsl:template match="/">
        <xsl:apply-templates select="$Protos/*"/>
    </xsl:template>
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <!--*                  This template copies the XML file verbatim                  *-->
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <xsl:template match="*">
        <xsl:copy>
            <xsl:for-each select="@*">
                <xsl:copy/>
            </xsl:for-each>
            <xsl:apply-templates/>
        </xsl:copy>
    </xsl:template>
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <xsl:template match="@*">
        <xsl:copy/>
    </xsl:template>
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <!--*              This template catches the Icon ProtoInstance              *-->
    <!--*              and makes copies of it using the data from the airfield xml  *-->
    <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
    <xsl:template match="ProtoInstance">
        <xsl:choose>
            <xsl:when test="@name='Dome'">
                <xsl:apply-templates select="$Data/root/*[@TARGET_TYPE='Populated']">
                    <xsl:with-param name="ProtoInstance" select="."/>
                </xsl:apply-templates>
                <xsl:apply-templates select="$Data/root/*[@TARGET_TYPE='Island']">
                    <xsl:with-param name="ProtoInstance" select="."/>
                </xsl:apply-templates>
<!--            <xsl:apply-templates select="$Data/root/IRAQ_CULTURAL">
                    <xsl:with-param name="ProtoInstance" select="."/>
                </xsl:apply-templates>
                <xsl:apply-templates select="$Data/root/IRAQ_LAND">
                    <xsl:with-param name="ProtoInstance" select="."/>
                </xsl:apply-templates>-->
            </xsl:when>
            <xsl:when test="@name='Icon'">
                <xsl:apply-templates          select="$Data/root/*[not(@TARGET_TYPE='Populated'          or
@TARGET_TYPE='Island')]">
                    <xsl:with-param name="ProtoInstance" select="."/>
```

```
                        </xsl:apply-templates>

                    <!-- <xsl:apply-templates select="$Data/root/IRAQ_DRAINAGE">
                            <xsl:with-param name="ProtoInstance" select="."/>
                        </xsl:apply-templates>
                        <xsl:apply-templates select="$Data/root/IRAQ_DRAINSUPP">
                            <xsl:with-param name="ProtoInstance" select="."/>
                        </xsl:apply-templates>
                        <xsl:apply-templates select="$Data/root/IRAQ_OCEAN">
                            <xsl:with-param name="ProtoInstance" select="."/>
                        </xsl:apply-templates>-->
                    </xsl:when>
                </xsl:choose>
            </xsl:template>
            <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
            <!--*    This template operates on table entries                                *-->
            <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
            <xsl:template match="root/*">
                <xsl:param name="ProtoInstance"/>
                <xsl:variable name="Coords">
                    <xsl:value-of select="@LAT"/>
                    <xsl:text> </xsl:text>
                    <xsl:value-of select="@LONG"/>
                    <xsl:text> </xsl:text>
                    <xsl:value-of select="@ELEV"/>
                </xsl:variable>
                <xsl:variable name="Table" select="normalize-space(name())"/>
                <xsl:variable name="Color" select="normalize-space(@*[contains(name(),'_COLOR')])"/>
                <xsl:variable name="IdField" select="name(@*[contains(name(),'_ID')])"/>
                <xsl:variable name="Id" select="normalize-space(@*[contains(name(),'_ID')])"/>
                <xsl:variable name="Image" select="normalize-space(@JPG)"/>
                <xsl:for-each select="$ProtoInstance/.">
                    <xsl:copy>
                        <xsl:for-each select="@*">
                            <xsl:copy/>
                        </xsl:for-each>
                        <xsl:apply-templates select="fieldValue[contains(@name,'Coords')]">
                            <xsl:with-param name="field" select="$Coords"/>
                        </xsl:apply-templates>
                        <xsl:apply-templates select="fieldValue[@name='iconColor']">
                            <xsl:with-param name="field" select="$Color"/>
                        </xsl:apply-templates>
                        <xsl:apply-templates select="fieldValue[contains(@name,'QueryUrl')]">
                            <xsl:with-param name="table" select="$Table"/>
                            <xsl:with-param name="idfield" select="$IdField"/>
                            <xsl:with-param name="idval" select="$Id"/>
                        </xsl:apply-templates>
                        <xsl:apply-templates select="fieldValue[@name='imageUrl']">
                            <xsl:with-param name="field" select="$Image"/>
                        </xsl:apply-templates>
                    </xsl:copy>
                </xsl:for-each>
            </xsl:template>
            <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
            <!--*    This template catches the fieldValue queryUrl and sets its value to    *-->
            <!--*    that of the field                                                   *-->
            <!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
            <xsl:template match="fieldValue[contains(@name,'QueryUrl')]">
                <xsl:param name="table"/>
                <xsl:param name="idfield"/>
                <xsl:param name="idval"/>
                <xsl:copy>
```

```
        <xsl:copy-of select="@name"/>
        <xsl:attribute name="value">
            <xsl:text>http://localhost/xmlvirBDV?sql=select+*+from+</xsl:text>
            <xsl:value-of select="$table"/>
            <xsl:text>+where+</xsl:text>
            <xsl:value-of select="$idfield"/>
            <xsl:text>=</xsl:text>
            <xsl:value-of select="concat($idval,'+FOR+XML+auto &amp; root = root')"/>
        </xsl:attribute>
        <xsl:apply-templates/>
    </xsl:copy>
</xsl:template>

<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<!--*     This template catches the fieldValue iconColor and sets its value    *-->
<!--*     that of the field                                                  *-->
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="fieldValue[contains(@name,'Color')]">
    <xsl:param name="field"/>
    <xsl:copy>
        <xsl:copy-of select="@name"/>
        <xsl:attribute name="value">
            <xsl:value-of select="$field"/>
        </xsl:attribute>
        <xsl:apply-templates/>
    </xsl:copy>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<!--*     This template catches the fieldValue imageUrland sets its value   *-->
<!--*     that of the field                                                   *-->
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="fieldValue[@name='imageUrl']">
    <xsl:param name="field"/>
    <xsl:copy>
        <xsl:copy-of select="@name"/>
        <xsl:attribute name="value">
            <xsl:value-of select="$field"/>
        </xsl:attribute>
        <xsl:apply-templates/>
    </xsl:copy>
</xsl:template>
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<!--*     This template catches the fieldValue domeCoords     and sets its value    *-->
<!--*     that of the field                                                   *-->
<!--*_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-*-->
<xsl:template match="fieldValue[contains(@name,'Coords')]">
    <xsl:param name="field"/>
    <xsl:copy>
        <xsl:copy-of select="@name"/>
        <xsl:attribute name="value">
            <xsl:value-of select="$field"/>
        </xsl:attribute>
        <xsl:apply-templates/>
    </xsl:copy>
</xsl:template>

      </xsl:stylesheet>
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F. X3D PROTOTYPE CODE FOR ICOSN THAT REPRESENT THE DATA FROM THE RELATIONAL DATABASE

This is the X3D code for the icons used in the Iraq scene. The icons are set up as prototypes that the XSLT file uses. Data from the query results XML file is used to fill in required fields in the icons.

```
<X3D>
    <head>
        <!--Import note:  the following meta tags were created during
    Vrml97ToX3d translation.  Please update or delete them as
    appropriate.-->
        <meta content="Icon.x3d" name="ICON"/>
        <meta content="*prototype for Icons and Domes*" name="description"/>
        <meta content="*Claude Hutton*" name="author"/>
        <meta content="Xeena VRML importer" name="translator"/>
        <meta content="*August 2003*" name="created"/>
        <meta content="09 May 2003" name="imported"/>
        <meta content="09 May 2003" name="revised"/>
        <meta content="*enter version here*" name="version"/>
        <meta content="*enter reference citation or relative/online url here*" name="reference"/>
        <meta content="*enter additional url/bibliographic reference information here*" name="reference"/>
        <meta content="*enter copyright information here* Example: Copyright (c) Web3D Consortium Inc.
2001" name="copyright"/>
        <meta content="*enter drawing filename/url here*" name="drawing"/>
        <meta content="*enter image filename/url here*" name="image"/>
        <meta content="*enter movie filename/url here*" name="movie"/>
        <meta content="*enter photo filename/url here*" name="photo"/>
        <meta content="*enter keywords here*" name="keywords"/>
        <meta content="*enter online url address for this file here*" name="url"/>
        <meta content="X3D-Edit, http://www.web3D.org/TaskGroups/x3d/translation/README.X3D-
Edit.html" name="generator"/>
        <meta content="Vrml97ToX3dNist, http://ovrt.nist.gov/v2_x3d.html" name="generator"/>
    </head>
    <Scene>
        <ProtoDeclare name="Icon">
            <ProtoInterface>
                <field accessType="initializeOnly" name="imageUrl" type="MFString" value=""/>
                <field accessType="initializeOnly" name="IconQueryUrl" type="MFString" value=""/>
                <field accessType="initializeOnly" name="iconColor" type="SFColor" value="1.0 1.0 1.0"/>
                <field accessType="initializeOnly" name="iconCoords" type="SFString"/>
            </ProtoInterface>
            <ProtoBody>
                <GeoLocation DEF="iconCoord">
                    <IS>
                        <connect nodeField="geoCoords" protoField="iconCoords"/>
                    </IS>
                    <GeoOrigin DEF="iconORIGIN" geoCoords="28.0 39.0 0" geoSystem="&quot;GD&quot;
&quot;WE&quot;" rotateYUp="false"/>
                    <Group>
                        <Anchor DEF="Icon" description="Pop up a new world" parameter="target=Data"
url="&#10;">
                            <IS>
                                <connect nodeField="url" protoField="IconQueryUrl"/>
                            </IS>
                            <LOD range="10000 50000">
                                <Transform scale="10.0 10.0 10.0">
```

```
                                    <Billboard>
                                        <Transform rotation="1.0 0.0 0.0 1.57" translation="0.0 13.0 0.0">
                                            <Shape>
                                                <Appearance>
                                                    <Material DEF="color" diffuseColor="0.6 0.6 0.6">
                                                        <IS>
                                                            <connect nodeField="diffuseColor"
protoField="iconColor"/>
                                                        </IS>
                                                    </Material>
                                                    <ImageTexture DEF="image"
url="&quot;Airfield.jpg&quot;&#10;">
                                                        <IS>
                                                            <connect nodeField="url" protoField="imageUrl"/>
                                                        </IS>
                                                    </ImageTexture>
                                                </Appearance>
                                                <Box size="8.0 0.1 8.0"/>
                                            </Shape>
                                        </Transform>
                                    </Billboard>
                                    <Transform DEF="POLE" translation="0.0 5.0 0.0">
                                        <Shape>
                                            <Appearance>
                                                <Material USE="color"/>
                                            </Appearance>
                                            <Cylinder height="8.0" radius="0.2"/>
                                        </Shape>
                                    </Transform>
                                    <Transform DEF="POLE_BASE" rotation="1.0 0.0 0.0 3.14">
                                        <Shape>
                                            <Appearance>
                                                <Material USE="color"/>
                                            </Appearance>
                                            <Cone bottomRadius="0.5"/>
                                        </Shape>
                                    </Transform>
                                </Transform>
                                <Transform scale="100.0 100.0 100.0">
                                    <Billboard>
                                        <Transform rotation="1.0 0.0 0.0 1.57" translation="0.0 13.0 0.0">
                                            <Shape>
                                                <Appearance>
                                                    <Material USE="color"/>
                                                    <ImageTexture USE="image"/>
                                                </Appearance>
                                                <Box size="8.0 0.1 8.0"/>
                                            </Shape>
                                        </Transform>
                                    </Billboard>
                                    <Transform USE="POLE"/>
                                    <Transform USE="POLE_BASE"/>
                                </Transform>
                                <Transform scale="1000.0 1000.0 1000.0">
                                    <Billboard>
                                        <Transform rotation="1.0 0.0 0.0 1.57" translation="0.0 13.0 0.0">
                                            <Shape>
                                                <Appearance>
                                                    <Material USE="color"/>
                                                    <ImageTexture USE="image"/>
                                                </Appearance>
                                                <Box size="8.0 0.1 8.0"/>
                                            </Shape>
                                        </Transform>
                                    </Billboard>
                                    <Transform USE="POLE"/>
                                    <Transform USE="POLE_BASE"/>
                                </Transform>
                            </LOD>
                        </Anchor>
                    </Group>
                </GeoLocation>
            </ProtoBody>
        </ProtoDeclare>
        <ProtoDeclare name="Dome">
            <ProtoInterface>
                <field accessType="initializeOnly" name="domeQueryUrl" type="MFString" value=""/>
                <field accessType="initializeOnly" name="domeColor" type="SFColor" value="1.0 1.0 1.0"/>
                <field accessType="initializeOnly" name="domeCoords" type="SFString"/>
            </ProtoInterface>
            <ProtoBody>
                <GeoLocation DEF="domeCoord">
                    <IS>
                        <connect nodeField="geoCoords" protoField="domeCoords"/>
                    </IS>
                    <GeoOrigin DEF="domeORIGIN" geoCoords="28.0 39.0 0"
geoSystem="&quot;GD&quot; &quot;WE&quot;" rotateYUp="false"/>
                    <Group>
                        <Anchor DEF="Dome" description="Pop up a new world" parameter="target=Data">
                            <IS>
                                <connect nodeField="url" protoField="domeQueryUrl"/>
                            </IS>
                            <LOD range="10000.0 50000.0">
                                <Transform scale="10 10 10">
                                    <Group DEF="dome">
                                        <Shape>
                                            <Appearance>
                                                <Material ambientIntensity="0.372549" diffuseColor="1.0 0.0
0.0" emissiveColor="1.0 0.0 0.0" transparency="0.4">
                                                    <IS>
                                                        <connect nodeField="diffuseColor"
protoField="domeColor"/>
                                                    </IS>
                                                </Material>
                                            </Appearance>
                                            <IndexedFaceSet coordIndex="0 1 2 -1 1 3&#10;4 -1 2 4 5 -
1&#10;3 6 7 -1 4 7&#10;8 -1 5 8 9 -1&#10;6 10 11 -1 7 11&#10;12 -1 8 12 13 -1&#10;9 13 14 -1 10
15&#10;16 -1 11 16 17 -1&#10;12 17 18 -1 13 18&#10;19 -1 14 19 20 -1&#10;1 4 2 -1 3 7&#10;4 -1 4 8 5 -
1&#10;6 11 7 -1 7 12&#10;8 -1 8 13 9 -1&#10;10 16 11 -1 11 17&#10;12 -1 12 18 13 -1&#10;13 19 14 -1 21
22&#10;23 -1 22 24 25 -1&#10;23 25 26 -1 24 27&#10;28 -1 25 28 29 -1&#10;26 29 30 -1 27 31&#10;32 -1
28 32 33 -1&#10;29 33 34 -1 30 34&#10;35 -1 31 0 2 -1&#10;32 2 5 -1 33 5&#10;9 -1 34 9 14 -1&#10;35 14
20 -1 22 25&#10;23 -1 24 28 25 -1&#10;25 29 26 -1 27 32&#10;28 -1 28 33 29 -1&#10;29 34 30 -1 31
2&#10;32 -1 32 5 33 -1&#10;33 9 34 -1 34 14&#10;35 -1 21 36 22 -1&#10;36 37 38 -1 22 38&#10;24 -1 37
39 40 -1&#10;38 40 41 -1 24 41&#10;27 -1 41 42 43 -1&#10;27 43 31 -1 31 44&#10;0 -1 36 38 22 -
1&#10;37 40 38 -1 38 41&#10;24 -1 40 42 41 -1&#10;41 43 27 -1 43 44&#10;31 -1 15 45 16 -1&#10;45 46
47 -1 16 47&#10;17 -1 46 48 49 -1&#10;47 49 50 -1 17 50&#10;18 -1 48 51 52 -1&#10;49 52 53 -1 50
53&#10;54 -1 18 54 19 -1&#10;51 55 56 -1 52 56&#10;57 -1 53 57 58 -1&#10;54 58 59 -1 19 59&#10;20 -1
45 47 16 -1&#10;46 49 47 -1 47 50&#10;17 -1 48 52 49 -1&#10;49 53 50 -1 50 54&#10;18 -1 51 56 52 -
1&#10;52 57 53 -1 53 58&#10;54 -1 54 59 19 -1&#10;15 60 45 -1 45 61&#10;46 -1 61 62 63 -1&#10;46 63
48 -1 63 64&#10;65 -1 48 65 51 -1&#10;64 66 67 -1 65 67&#10;68 -1 51 68 55 -1&#10;60 61 45 -1 61
63&#10;46 -1 62 64 63 -1&#10;63 65 48 -1 64 67&#10;65 -1 65 68 51 -1&#10;55 69 56 -1 69 70&#10;71 -1
56 71 57 -1&#10;70 72 73 -1 71 73&#10;74 -1 57 74 58 -1&#10;72 75 76 -1 73 76&#10;77 -1 74 77 78 -
1&#10;58 78 59 -1 75 79&#10;80 -1 76 80 81 -1&#10;77 81 82 -1 78 82&#10;83 -1 59 83 20 -1&#10;69 71
56 -1 70 73&#10;71 -1 71 74 57 -1&#10;72 76 73 -1 73 77&#10;74 -1 74 78 58 -1&#10;75 80 76 -1 76
```

```
81&#10;77 -1 77 82 78 -1&#10;78 83 59 -1 55 84&#10;69 -1 84 85 86 -1&#10;69 86 70 -1 85 87&#10;88 -1
86 88 89 -1&#10;70 89 72 -1 87 90&#10;91 -1 88 91 92 -1&#10;89 92 93 -1 72 93&#10;75 -1 90 94 95 -
1&#10;91 95 96 -1 92 96&#10;97 -1 93 97 98 -1&#10;75 98 79 -1 84 86&#10;69 -1 85 88 86 -1&#10;86 89
70 -1 87 91&#10;88 -1 88 92 89 -1&#10;89 93 72 -1 90 95&#10;91 -1 91 96 92 -1&#10;92 97 93 -1 93
98&#10;75 -1 79 99 80 -1&#10;99 100 101 -1 80 101&#10;81 -1 100 102 103 -1&#10;101 103 104 -1 81
104&#10;82 -1 102 105 106 -1&#10;103 106 107 -1 104 107&#10;108 -1 82 108 83 -1&#10;105 21 23 -1
106 23&#10;26 -1 107 26 30 -1&#10;108 30 35 -1 83 35&#10;20 -1 99 101 80 -1&#10;100 103 101 -1 101
104&#10;81 -1 102 106 103 -1&#10;103 107 104 -1 104 108&#10;82 -1 105 23 106 -1&#10;106 26 107 -1
107 30&#10;108 -1 108 35 83 -1&#10;79 109 99 -1 109 110&#10;111 -1 99 111 100 -1&#10;110 112 113 -1
111 113&#10;114 -1 100 114 102 -1&#10;112 115 116 -1 113 116&#10;117 -1 114 117 118 -1&#10;102 118
105 -1 115 119&#10;120 -1 116 120 121 -1&#10;117 121 122 -1 118 122&#10;123 -1 105 123 21 -
1&#10;109 111 99 -1 110 113&#10;111 -1 111 114 100 -1&#10;112 116 113 -1 113 117&#10;114 -1 114
118 102 -1&#10;115 120 116 -1 116 121&#10;117 -1 117 122 118 -1&#10;118 123 105 -1 119 115&#10;124
-1 115 112 128 -1&#10;124 128 125 -1 112 110&#10;129 -1 128 129 130 -1&#10;125 130 126 -1 110
109&#10;131 -1 129 131 132 -1&#10;130 132 133 -1 126 133&#10;127 -1 109 79 98 -1&#10;131 98 97 -1
132 97&#10;96 -1 133 96 95 -1&#10;127 95 94 -1 115 128&#10;124 -1 112 129 128 -1&#10;128 130 125 -1
110 131&#10;129 -1 129 132 130 -1&#10;130 133 126 -1 109 98&#10;131 -1 131 97 132 -1&#10;132 96
133 -1 133 95&#10;127 -1 39 37 135 -1&#10;37 36 137 -1 135 137&#10;138 -1 136 138 139 -1&#10;36 21
123 -1 137 123&#10;122 -1 138 122 121 -1&#10;139 121 120 -1 134 120&#10;119 -1 37 137 135 -
1&#10;135 138 136 -1 36 123&#10;137 -1 137 122 138 -1&#10;138 121 139 -1 139 120&#10;134 -1 94 90
140 -1&#10;90 87 141 -1 87 85&#10;142 -1 141 142 143 -1&#10;85 84 144 -1 142 144&#10;145 -1 84 55
68 -1&#10;144 68 67 -1 145 67&#10;66 -1 90 141 140 -1&#10;87 142 141 -1 85 144&#10;142 -1 142 145
143 -1&#10;84 68 144 -1 144 67&#10;145 -1" creaseAngle="0.01" solid="false">
                                                                <Coordinate point="0.5257 0.0 0.8507&#10;0.3477 0.0
0.9376&#10;0.4636 0.1875 0.866&#10;0.1227 0.0 0.9924&#10;0.2531 0.2047 0.9455&#10;0.368 0.397
0.8408&#10;-0.1227 0.0 0.9924&#10;0.0 0.2116 0.9773&#10;0.1308 0.4233 0.8965&#10;0.2453 0.5955
0.765&#10;-0.3477 0.0 0.9376&#10;-0.2531 0.2047 0.9455&#10;-0.1308 0.4233 0.8965&#10;0.0 0.6142
0.7891&#10;0.1159 0.7501 0.6511&#10;-0.5257 0.0 0.8507&#10;-0.4636 0.1875 0.866&#10;-0.368 0.397
0.8408&#10;-0.2453 0.5955 0.765&#10;-0.1159 0.7501 0.6511&#10;0.0 0.8507 0.5257&#10;0.8507 0.5257
0.0&#10;0.866 0.4636 0.1875&#10;0.7501 0.6511 0.1159&#10;0.8408 0.368 0.397&#10;0.7408 0.5844
0.3313&#10;0.5955 0.765 0.2453&#10;0.765 0.2453 0.5955&#10;0.6849 0.4732 0.5541&#10;0.5541 0.6849
0.4732&#10;0.397 0.8408 0.368&#10;0.6511 0.1159 0.7501&#10;0.5844 0.3313 0.7408&#10;0.4732 0.5541
0.6849&#10;0.3313 0.7408 0.5844&#10;0.1875 0.866 0.4636&#10;0.9376 0.3477 0.0&#10;0.9924 0.1227
0.0&#10;0.9455 0.2531 0.2047&#10;0.9924 0.0 0.0&#10;0.9773 0.0 0.2116&#10;0.8965 0.1308
0.4233&#10;0.8965 0.0 0.4233&#10;0.7891 0.0 0.6142&#10;0.6511 0.0 0.7501&#10;-0.6511 0.1159
0.7501&#10;-0.765 0.2453 0.5955&#10;-0.5844 0.3313 0.7408&#10;-0.8408 0.368 0.397&#10;-0.6849
0.4732 0.5541&#10;-0.4732 0.5541 0.6849&#10;-0.866 0.4636 0.1875&#10;-0.7408 0.5844 0.3313&#10;-
0.5541 0.6849 0.4732&#10;-0.3313 0.7408 0.5844&#10;-0.8507 0.5257 0.0&#10;-0.7501 0.6511
0.1159&#10;-0.5955 0.765 0.2453&#10;-0.397 0.8408 0.368&#10;-0.1875 0.866 0.4636&#10;-0.6511 0.0
0.7501&#10;-0.7891 0.0 0.6142&#10;-0.8965 0.0 0.4233&#10;-0.8965 0.1308 0.4233&#10;-0.9773 0.0
0.2116&#10;-0.9455 0.2531 0.2047&#10;-0.9924 0.0 0.0&#10;-0.9924 0.1227 0.0&#10;-0.9376 0.3477
0.0&#10;-0.7501 0.6511 -0.1159&#10;-0.5955 0.765 -0.2453&#10;-0.6142 0.7891 0.0&#10;-0.397 0.8408 -
0.368&#10;-0.4233 0.8965 -0.1308&#10;-0.4233 0.8965 0.1308&#10;-0.1875 0.866 -0.4636&#10;-0.2047
0.9455 -0.2531&#10;-0.2116 0.9773 0.0&#10;-0.2047 0.9455 0.2531&#10;0.0 0.8507 -0.5257&#10;0.0
0.9376 -0.3477&#10;0.0 0.9924 -0.1227&#10;0.0 0.9924 0.1227&#10;0.0 0.9376 0.3477&#10;-0.866 0.4636
-0.1875&#10;-0.8408 0.368 -0.397&#10;-0.7408 0.5844 -0.3313&#10;-0.765 0.2453 -0.5955&#10;-0.6849
0.4732 -0.5541&#10;-0.5541 0.6849 -0.4732&#10;-0.6511 0.1159 -0.7501&#10;-0.5844 0.3313 -
0.7408&#10;-0.4732 0.5541 -0.6849&#10;-0.3313 0.7408 -0.5844&#10;-0.5257 0.0 -0.8507&#10;-0.4636
0.1875 -0.866&#10;-0.368 0.397 -0.8408&#10;-0.2453 0.5955 -0.765&#10;-0.1159 0.7501 -
0.6511&#10;0.1875 0.866 -0.4636&#10;0.397 0.8408 -0.368&#10;0.2047 0.9455 -0.2531&#10;0.5955 0.765
-0.2453&#10;0.4233 0.8965 -0.1308&#10;0.2116 0.9773 0.0&#10;0.7501 0.6511 -0.1159&#10;0.6142
0.7891 0.0&#10;0.4233 0.8965 0.1308&#10;0.2047 0.9455 0.2531&#10;0.1159 0.7501 -0.6511&#10;0.2453
0.5955 -0.765&#10;0.3313 0.7408 -0.5844&#10;0.368 0.397 -0.8408&#10;0.4732 0.5541 -
0.6849&#10;0.5541 0.6849 -0.4732&#10;0.4636 0.1875 -0.866&#10;0.5844 0.3313 -0.7408&#10;0.6849
0.4732 -0.5541&#10;0.7408 0.5844 -0.3313&#10;0.5257 0.0 -0.8507&#10;0.6511 0.1159 -
0.7501&#10;0.765 0.2453 -0.5955&#10;0.8408 0.368 -0.397&#10;0.866 0.4636 -0.1875&#10;0.3477 0.0 -
0.9376&#10;0.1227 0.0 -0.9924&#10;-0.1227 0.0 -0.9924&#10;-0.3477 0.0 -0.9376&#10;0.2531 0.2047 -
0.9455&#10;0.1308 0.4233 -0.8965&#10;0.0 0.2116 -0.9773&#10;0.0 0.6142 -0.7891&#10;-0.1308 0.4233 -
0.8965&#10;-0.2531 0.2047 -0.9455&#10;0.6511 0.0 -0.7501&#10;0.9773 0.0 -0.2116&#10;0.8965 0.0 -
0.4233&#10;0.9455 0.2531 -0.2047&#10;0.8965 0.1308 -0.4233&#10;0.7891 0.0 -0.6142&#10;-0.6511 0.0 -
0.7501&#10;-0.7891 0.0 -0.6142&#10;-0.8965 0.1308 -0.4233&#10;-0.8965 0.0 -0.4233&#10;-0.9455
0.2531 -0.2047&#10;-0.9773 0.0 -0.2116&#10;"/>
```

```
                                            </IndexedFaceSet>
                                        </Shape>
                                    </Group>
                                </Transform>
                                <Transform scale="100.0 100.0 100.0">
                                    <Group USE="dome"/>
                                </Transform>
                                <Transform scale="1000.0 1000.0 1000.0">
                                    <Group USE="dome"/>
                                </Transform>
                            </LOD>
                        </Anchor>
                    </Group>
                </GeoLocation>
            </ProtoBody>
        </ProtoDeclare>
        <ProtoInstance name="Icon">
            <fieldValue name="IconQueryUrl"
value="http://localhost/xmlvirBDV?sql=select+*+from+IRAQ_AIRFEILDS+FOR+XML+auto&amp;root=root"/
>
            <fieldValue name="imageUrl" value="Airfield.jpg"/>
            <fieldValue name="iconColor" value="0.6 0.6 0.6"/>
            <fieldValue name="iconCoords" value="33.3328 44.3639 0"/>
        </ProtoInstance>
        <ProtoInstance name="Dome">
            <fieldValue name="domeQueryUrl"
value="http://localhost/xmlvirBDV?sql=select+*+from+IRAQ_AIRFEILDS+FOR+XML+auto&amp;root=root"/
>
            <fieldValue name="domeColor" value="0.6 0.6 0.6"/>
            <fieldValue name="domeCoords" value="33.3328 44.3639 0"/>
        </ProtoInstance>
    </Scene>
</X3D>
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[Blais 2001] Blais, Curtis; Brutzman, Don; Horner, Doug; Nicklaus, Shane, Major USMC. "WEB-BASED 3D TECHNOLOGY FOR SCENARIO AUTHORING AND VISUALIZATION: THE SAVAGE PROJECT." I/ITSEC Proceedings CD. 2001.

[DEFENSE 1999] Department of Defense; MIL-STD-2525B, 30 January 1999.

[DTED 1996] Performance Specification, Digital Terrain Elevation Data (DTED), 19 April 1996, MIL-PRF—89020A, Defense Mapping Agency, Fairfax, Virginia, US.

[Harney 2003] Harney, James. Amalyzing Tactical Effectiveness for Anti-Terrorist Force Protection (AT/FP) Using Using X3D Graphics and Agent-Based Simulation, Master's Thesis, Naval Postgraduate School. Monterey, California, March 2003.

[Hiles 2003] Hiles, John. "Integrated Asymmetric Goal Organization (IAGO): A Multiagent Model of Conceptual Blending". White paper written at NPS. 2003.

[Hunter 2001] Hunter, David; Cagle, Kurt; Dix, Chris; Kovack, Roger; Pinnock, Jonathan, Rafter, Jeff; Beginning XML (2nd Edition), 2001, Wrox Press, Birmingham, United Kingdom.

[Kay 2001] Kay, Michael, *XSLT Programmer's Reference 2nd Edition,* Wrox Press, 2001, Birmingham, United Kingdom.

[Koester 2003] Koester, Michael, COP Lab. Power Point brief. [http://jitc.fhu.disa.mil/gccsiop/briefs.htm]. May 2003.

[NATO 2002] NATO, "Overview of the Land C2 Information Exchange Data Model (LC2IEDM)", 18 March 2000.

[Maranne 2003] Maranne, Eric. Equip@ge 3D Information Systems. www.geovrml.com. August 2003.

[Nicklaus 2001] Nicklaus, S. (2001). Scenario Authoring and Visualization for Advanced Graphical Environments. Master's Thesis, Naval Postgraduate School.

[Neushul 2003] Neushul, James (2003) Draft Master's Thesis, Naval

Postgraduate School.

[Onley 2003] Onley, Dawn; Menke, Susan. "DOD seeks to exploit intelligence". Government Computer News. 23 June. Pp. 1, 10. 2003.

[Roehl 1997] Bernie, Roehl; Couch, Justin; Reed-BallReich, Cindy; Rohaly, Tim; Brown, Geoff; LATE NIGHT VRML 2.0 with Java, 1997, Ziff-Davis Press, an imprint of Macmillan Computer Publishing, Emeryville, California

[Savage 2003] NPS, Scenario Authoring and Visualization for Advanced Graphical Environments (SAVAGE), http://web.nps.navy.mil/~brutzman/Savage/contents.html (accessed August 2003), August 2003.

[Wagner 2002] Au, Wagner James. "Weapons of mass distraction." [http://www.salon.com/tech/feature/2002/10/04/why_we_fight/print.html]. October 2002.

[XMSF 2002] Brutzman, D., Morse, K., Pullen, M., Zyda, M., "XMSF 2002 Finding and Recommendations Report", Technical Challenges Workshop and Strategic

Opportunities Symposium, 22 October 2002. Available at: http://www.movesinstitute.org/private/tmp/501/Temporary%20Items/Library/Apache2/ht

docs/xmsf/XmsfWorkshopSymposiumReportOctober2002.pdf (accessed August 2003)

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Brigadier General Michael E. Ennis
   Director, Intelligence HQMC
   HQ USMC, Department I
   Washington, DC

4. Colonel Joseph A Bruder IV
   MARCORSYSCOM
   Quantico, Va

5. Lieutenant Colonel David G Gran
   MARCORSYSCOM
   Quantico, Va

6. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

7. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

8. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

9. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

10. Director, Studies and Analysis Division, MCCDC, Code C45
    Quantico, Virginia

11. Mr. John Audia
    SPAWAR
    San Diego, California

12. Robert Eubank
    NSA
    Ft. Meade, Maryland

13. Dr. S. K. Numrich
Defense Modeling and Simulation Office (DMSO)
Alexandria, Virginia

14. Associate Professor Don Brutzman
Naval Postgraduate School
Monterey, California

15. Research Associate Curt Blais
Naval Postgraduate School
Monterey, California

16. Research Associate Jeff Weekley
Naval Postgraduate School
Monterey, California

17. Claude Hutton, CAPT, USMC
Naval Postgraduate School
Monterey, California

18. Jerry Kniphfer
Army model Exchange
Redstone Arsenal, Alabama